Санкт-Петербургский государственный университет

Степырев Даниил Федорович

Производственная практика

Извлечение данных SIM-карты с использованием считывателя карт

Научный руководитель:
доцент кафедры СП, к.т.н., Ю.В. Литвинов

Консультант:
архитектор ООО «Цифровая Корпоративная Защита» Н.М. Тимофеев

Санкт-Петербург
2024

Оглавление

1. Введение

С развитием технологий в современном мире возрастает и число цифровых преступлений [1]. Примерами таких преступлений могут быть кражи личных данных, распространение противоправной информации, вмешательство в работу приложений и многое другое.

Цифровая криминалистика — наука, помогающая обнаружить, зафиксировать и исследовать компьютерные доказательства для подтверждения противоправных действий. Цифровая криминалистика не предназначена для противоправных действий, она используется только с разрешения суда.

Для обнаружения доказательств, подтверждающих цифровые преступления, эксперты цифровой криминалистики снимают данные с различных устройств [2]. Полезные для расследований данные могут находиться в памяти цифровых устройств, внешних запоминающих устройств, в облачных хранилищах данных. Одним из таких устройств является разновидность смарт-карт — Сим-карта.

На Сим-карте хранится идентификационная информация о пользователе: международный номер мобильного абонента (IMSI), ключ аутентификации пользователя (KI). Однако помимо идентификационной информации на Сим-карте также хранятся и данные о пользователе: телефонная книга, журнал звонков, принятые и отправленные SMS-сообщения [3]. Подобная информация может быть полезна экспертам в области цифровой криминалистики.

Чтение Сим-карты производится с помощью специальных устройств, называемых считывателями карт [4]. Сим-карта вставляется в специальный разъём устройства, которое затем подключается через USB-порт к компьютеру.

Интерес к снятию данных с Сим-карт возник у компании «Цифровая корпоративная защита» при разработке продукта Belkasoft X[1]. Belkasoft X — инструмент цифровой криминалистики, разработанный для снятия и анализа данных с компьютера, мобильных устройств, об-

[1]https://belkasoft.com/ru/x (дата обращения: 23.12.22).

лачных хранилищ.

На момент написания данной работы существует несколько работающих проектов, поддерживающих извлечение данных Сим-карты с использованием считывателя карт. Тем не менее эти проекты либо не позволяют выполнять полное снятие и разбор файловой системы Сим-карты, либо представляют собой условно-бесплатные ограниченные версии.

Однако с помощью инструментов обратной разработки можно выяснить принцип снятия данных с Сим-карты и реализовать модуль, позволяющий извлекать всю файловую систему Сим-карты с использованием считывателя карт. Реализация такой функциональности для коммерческого продукта Belkasoft X и стала целью данной работы.

2. Постановка задачи

Целью представленной работы является разработка модуля, предназначенного для извлечения данных Сим-карты с использованием считывателя карт. Для достижения цели были поставлены задачи.

- Выполнить обзор предметной области — файловой системы Сим-карты, аналогов разрабатываемого модуля.
- Спроектировать и реализовать модуль, извлекающий файловую систему Сим-карты с использованием считывателя карт.
- Спроектировать и реализовать модуль, выполняющий разбор извлечённых данных Сим-карты.
- Выполнить интеграцию разработанного модуля в продукт Belkasoft X.

3. Обзор

3.1. Файловая система SIM-карты

Файловая система SIM-карты имеет древовидную структуру. Зачастую древовидная файловая система состоит из файлов и каталогов, позволяющих группировать файлы. Например, так устроена файловая система Linux [6]. В файловой системе SIM-карты каталог также является файлом.

Корневым элементом файловой системы SIM-карты является главный файл MF (Master File). Он содержит в себе все остальные файлы, хранимые на SIM-карте [5]. Помимо главного файла существует ещё два типа файлов: элементарные и вложенные.

Элементарные файлы EF (Elementary File) содержат в себе только данные. Такие файлы не могут содержать внутри себя другие файлы. Данные в элементарных файлах хранятся в виде байтов. Пример элементарного файла — файл EF_IMSI [7]. Этот файл содержит информацию об IMSI-номере SIM-карты.

Вложенные файлы DF (Dedicated File) содержат в себе другие файлы. Во вложенном файле могут находиться как элементарные файлы, так и другие вложенные файлы. Зачастую данные разбиваются на несколько элементарных файлов и хранятся в одном вложенном файле. Примером подобных файлов может являться телефонная книга номеров сокращённого набора ADN [8].

Устройство файловой системы SIM-карты представлено на рис. 1 (диаграмма компонент UML). На верхнем уровне файловой системы SIM-карты находится только корневой файл MF. Он содержит в себе вложенные файлы с описанием уровня GSM (файл DF_GSM) и уровня TELECOM (файл DF_TELECOM). Кроме вложенных файлов на MF уровне находится элементарный файл с описанием уникального серийного номера SIM-карты ICCID (файл EF_ICCID).

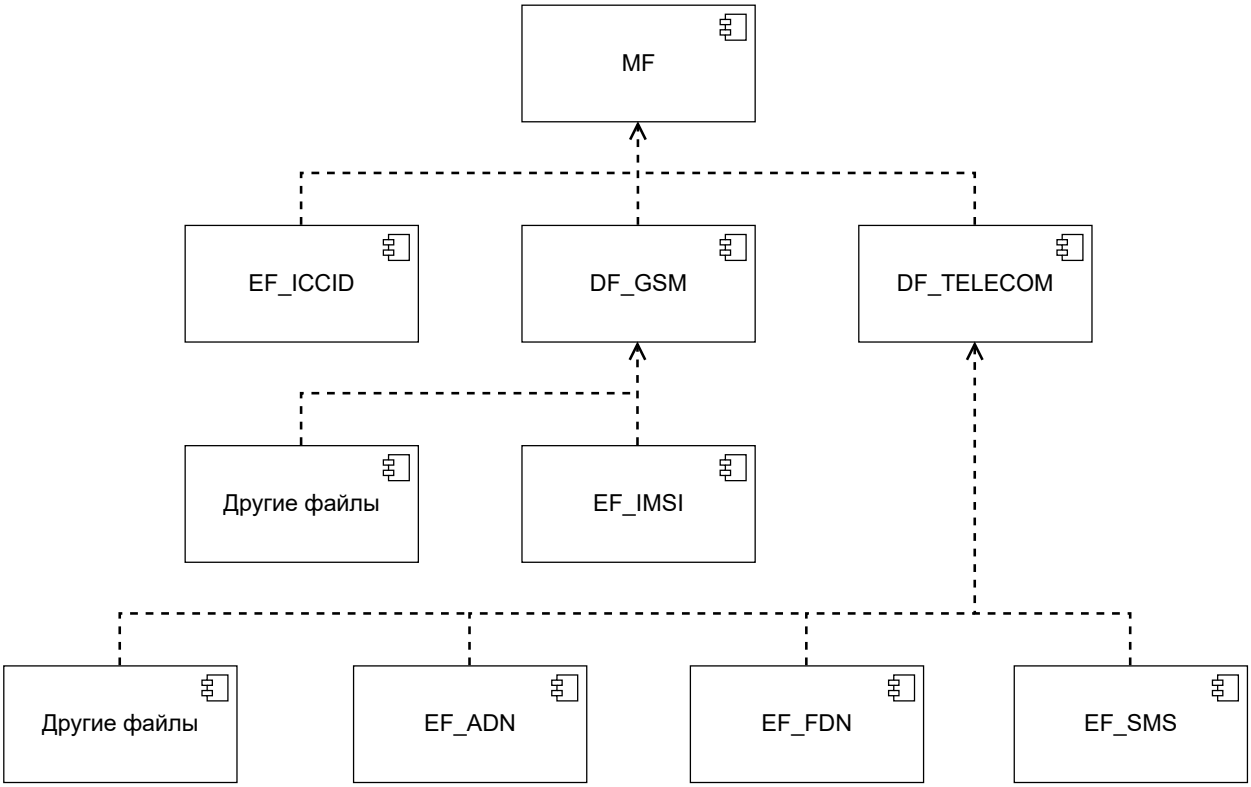

Рис. 1: Файловая система SIM-карты.

На уровне GSM находятся файлы, содержащие информацию, связанную с сетью. Также на уровне GSM располагается файл, хранящий международный идентификатор мобильного абонента IMSI (файл EF_IMSI).

На уровне TELECOM хранятся элементарные файлы с номерами сокращённого набора (файл EF_ADN), фиксированными номерами (файл EF_FDN), отправленными и полученными SMS-сообщениями (файл EF_SMS). На уровне TELECOM находятся другие элементарные файлы, содержащие конфигурационные параметры различных сервисов.

3.2. Считыватель карт

Считыватель карт — специальное устройство, предназначенное для взаимодействия с SIM-картой. Устройство имеет отдельное отверстие для подключения SIM-карты. Считыватель карт подключается через USB-порт компьютера и позволяет читать данные SIM-карты по COM-

порту. Пример считывателя карт, используемого в данной работе, представлен рис. 2.

Рис. 2: Изображение считывателя карт.

3.3. Обзор аналогов

В обзоре описаны популярные инструменты, предназначенные для извлечения данных SIM-карты с использованием считывателя карт. Аналоги выбирались с помощью поисковой системы Google с использование ключевых слов «SIM card», «acquisition», «Reader», «tools».

3.3.1. E3: Electronic Evidence Examine

E3: Electronic Evidence Examine — продукт, распространяемый компанией Paraben, позволяющий извлекать данные SIM-карты [9]. Инструмент ранее существовал как отдельный проект SIMCon, однако компания Paraben выкупила права на пользование и интегрировала его в качестве модуля продукта E3: Electronic Evidence Examiner. Инструмент позволяет извлекать и анализировать всю файловую систему SIM-карты, а также поддерживает верификацию PIN-кода. Проект поддерживается в настоящее время. Продукт E3: Electronic Evidence Examine платный, стоимость лицензии составляет 1895$ в год. Однако компания Paraben предоставляет бесплатную версию продукта на 30 дней.

3.3.2. Oxygen Forensics Detective

Oxygen Forensics Detective — продукт компании Oxygen Forensics, позволяющий извлекать данные SIM-карты [10]. Помимо извлечения и анализа всей файловой системы SIM-карты инструмент поддерживает верификацию PIN-кода. Проект поддерживается в настоящее время. Продукт платный, стоимость лицензии составляет 8090€ в год. Компания Oxygen Forensics предоставляет бесплатную версию продукта на 20 дней.

3.3.3. SimLAB

SimLAB — продукт с открытым исходным кодом, который позволяет извлекать файловую систему SIM-карты [11]. Инструмент также поддерживает верификацию PIN-кода. Извлечение данных происходит в бинарном формате без дальнейшего преобразования байтов в текст. Инструмент simLAB не позволяет разбирать извлечённые файлы SIM-карты. В описании simLAB также указано, что SIM-карта может быть заблокирована, а автор не даёт никакой гарантии. Последние обновления в проекте были в 2016 году.

3.3.4. Osmo-sim-auth

Osmo-sim-auth — продукт с открытым исходным кодом, позволяющий извлекать файловую систему SIM-карты [12]. Инструмент помимо извлечения файловой системы SIM-карты также поддерживает верификацию PIN-кода. Osmo-sim-auth не позволяет проанализировать извлечённые с SIM-карты файлы. Последние обновления в проекте были в 2017 году.

3.3.5. DualSIMCard

DualSIMCard — продукт с открытым исходным кодом, предназначенный для извлечения данных SIM-карты [13]. Инструмент позволяет извлекать лишь часть файлов SIM-карты и не извлекает файлы, свя-

занные с данными пользователя. DualSIMCard не поддерживает разбор извлечённых данных и верификацию PIN-кода. Последние обновления в проекте были в 2019 году.

3.3.6. Сравнение аналогов

Рассмотренные аналоги представлены в таблице 1. Большинство рассмотренных аналогов позволяет извлечь файловую систему SIM-карты. Однако не все из них поддерживают полный разбор извлечённой файловой системы. Только платные продукты E3: Electronic Evidence Examine [9] и Oxygen Forensics Detective [10] предоставляют такую возможность. Продукты simLAB, DualSIMCard [11, 13] позволяют только извлекать данные SIM-карты. С помощью доступных инструментов нельзя гарантированно извлечь и разобрать файловую систему SIM-карты.

Название	Извлечение файловой системы SIM-карты	Разбор файловой системы SIM-карты	Верификация PIN-кода	Актуальность	Доступность
E3: Electronic Evidence Examine	Есть	Есть	Есть	Поддерживается в настоящее время	Платная лицензия стоимостью 1895$ в год, триальная версия на 30 дней
Oxygen Forensics Detective	Есть	Есть	Есть	Поддерживается в настоящее время	Платная лицензия стоимостью 8090€ в год, триальная версия на 20 дней
SimLAB	Есть	Нет	Есть	Последнее обновление в 2016 году	В свободном доступе
Osmo-sim-auth	Есть	Нет	Есть	Последнее обновление в 2017 году	В свободном доступе
DualSIM Card	Только данные оператора	Нет	Нет	Последнее обновление в 2019 году	В свободном доступе

Таблица 1: Сравнительные характеристики аналогов

Продукты E3: Electronic Evidence Examine, Oxygen Forensics Detective [9, 10] поддерживаются в настоящее время. Последняя модификация в проектах SimLAB, Osmo-sim-auth, DualSIMCard [11,13] была в 2016, 2017 и 2019 годах.

Большинство продуктов поддерживают алгоритм верификации PIN-кода SIM-карты. Такая функциональность недоступна только в проекте DualSIMCard.

С помощью инструментов с открытым исходным кодом нельзя из-

влечь и разобрать всю файловую систему SIM-карты. Также с помощью продуктов SimLAB, Osmo-sim-auth, DualSimCard [11, 12, 13] не удалось извлечь данные с SIM-карты с использованием тестового считывателя данных.

3.4. Работа с SIM-картой через COM-порт

Для проверки работоспособности считывателя SIM-карты и проверки идей относительно принципов извлечения данных были применены средства обратной разработки. Был перехвачен трафик между устройством и одним из доступных продуктов. Для перехвата трафика использовалась бесплатная версия продукта Serial Port Monitor [14]. Архитектура решения представлена на рис. 3 (диаграмма последовательности UML).

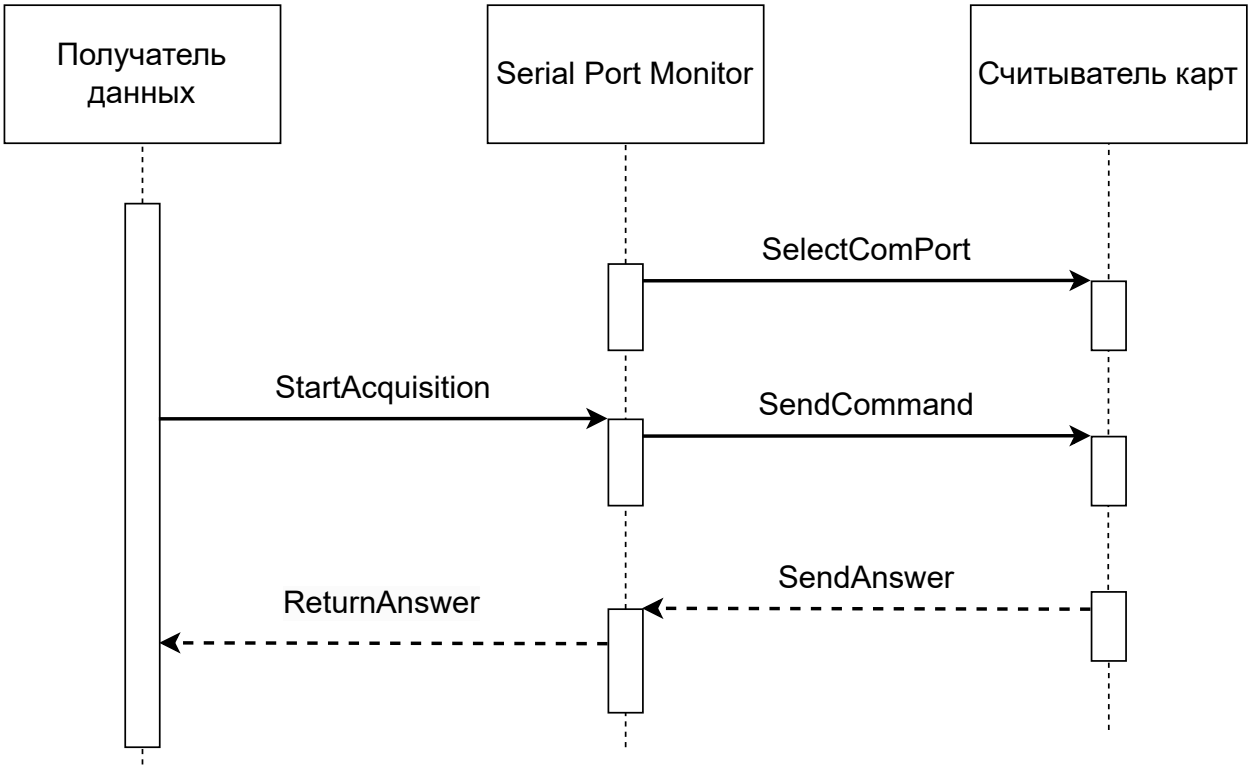

Рис. 3: Принципиальная схема перехвата команд.

Перед началом извлечения данных на порт считывателя карт устанавливается перехватчик Serial Port Monitor (сообщение SelectComPort от SerialPortMonitor к считывателю карт). Это позволяет перехватить

все отправляемые команды, а также получить ответы на них от считывателя карт.

Затем начинается извлечение данных SIM-карты. Отправляемые команды перехватываются Serial Port Monitor, записываются в файл и пересылаются считывателю карт (сообщение SendCommand от Serial Port Monitor к считывателю карт).

Полученные от считывателя карт ответы также перехватываются и записываются в файл (сообщение SendAnswer от считывателя карт к Serial Port Monitor). Затем перехватчик пересылает ответы получателю данных (сообщение ReturnAnswer от Serial Port Monitor к получателю данных).

3.5. Извлечение данных SIM-карты

В результате анализа удалось убедиться, что взаимодействие со считывателем карт выполняется с использованием стандарта смарт-карт ISO 7816 [15].

Для доступа к файлу SIM-карты необходимо перейти в директорию, в которой находится файл. Переход в директорию осуществляется с помощью команды выбора файла. Для доступа во вложенную директорию необходимо сначала перейти в родительскую директорию.

Извлечение данных с SIM-карты может быть ограничено установленным PIN-кодом. Определить, установлен ли такой код на SIM-карту, можно по полученному ответу от считывателя карт. Если PIN-код установлен, для доступа к данным SIM-карты необходимо выполнить его верификацию.

На ввод PIN-кода предоставляется три попытки. После трёх неправильных попыток ввода PIN[2]-кода SIM-карта переходит в режим ввода PUK[3]-кода. После десяти неправильных попыток ввода PUK-кода SIM-карта блокируется.

После считывания файла необходимо выполнить разбор полученно-

[2]PIN — Personal Identification Number

[3]PUK — Personal Unlocking Key

го ответа от считывателя карт. Необходимо удалить из ответа информацию об исполненной команде и дополненные байты. Оставшиеся байты необходимо разобрать согласно алгоритму кодирования, указанному в стандарте ISO 7816.

3.6. Подтверждение PIN-кода SIM-карты

Установленный на SIM-карту PIN-код препятствует извлечению файловой системы. При попытке извлечь данные SIM-карты с неподтверждённым PIN-кодом вернётся код ошибки.

Для определения наличия установленного PIN-кода необходимо отправить специальный запрос на SIM-карту. Ответ на этот запрос содержит информацию об установленном PIN-коде, а также число оставшихся попыток для его подтверждения. По умолчанию пользователю предоставляется три попытки ввода PIN-кода. После трёх неудачных попыток ввода PIN-кода на SIM-карту устанавливается PUK-код.

PUK-код – восьмизначный код, предоставляемый оператором сотовой связи при выпуске SIM-карты. Для подтверждения PUK-кода пользователю предоставляется десять попыток. После десяти неудачных вводов PUK-кода SIM-карта блокируется.

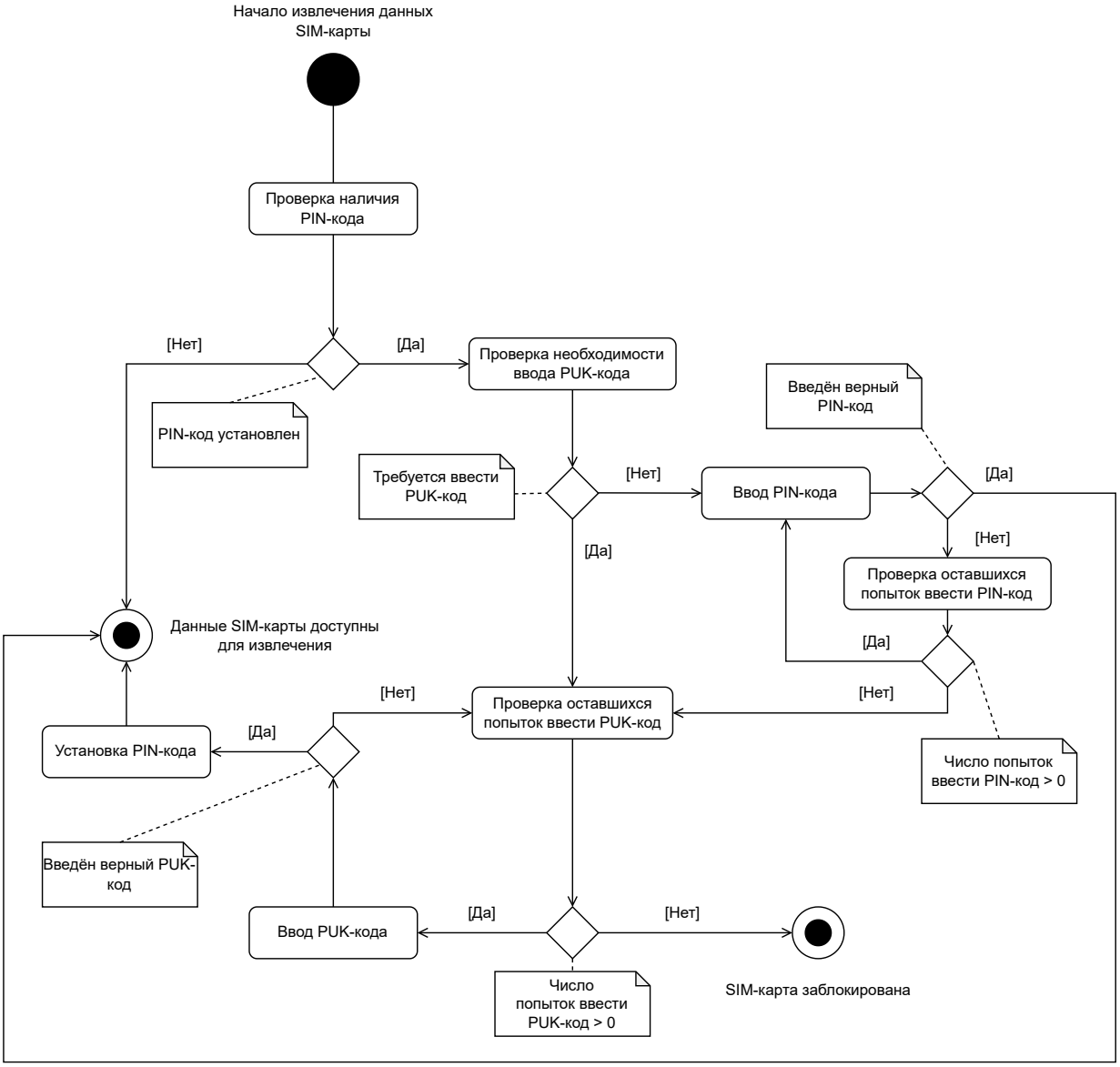

Рис. 4: Алгоритм потдверждения PIN-кода.

Алгоритм проверки и подтверждения PIN-кода SIM-карты представлен на рис. 4 (диаграмма активностей UML). Перед началом извлечения файловой системы SIM-карты выполняется проверка, установлен ли PIN-код. Для этого на SIM-карту отправляется специальный запрос. Из ответа на запрос можно выяснить наличие установленного кода, а также число оставшихся попыток ввода.

В случае, когда SIM-карта не требует подтверждения никакого кода, начинается извлечение файловой системы SIM-карты. Если требуется подтверждение кода, то по ответу на запрос определяется, какой код установлен: PIN- или PUK-код.

Если требуется ввести PIN-код, то перед извлечением файловой системы SIM-карты необходимо отправить запрос с правильным кодом. После каждого запроса приходит ответ с результатом проверки введённого кода. Если был введён правильный PIN-код, файловая система SIM-карты становится доступной для извлечения, число попыток ввода PIN-кода обновляется до трёх. Если был введён неверный PIN-код, число попыток ввода уменьшается на одну. Если на последней попытке был введён неверный PIN-код, на SIM-карту устанавливается PUK-код.

Для ввода PUK-кода предоставляется десять попыток. Алгоритм ввода и проверки PUK-кода аналогичен алгоритму подтверждения PIN-кода. Отличие состоит в том, что после ввода правильного PUK-кода, необходимо задать новое значение PIN-кода. По умолчанию в Belkasoft X устанавливается значение «0000». Если на последней попытке был введён неверный PUK-код, SIM-карта блокируется.

3.7. Разбор извлечённых данных SIM-карты

После извлечения данных SIM-карты получается набор файлов, структура которых повторяет устройство файловой системы SIM-карты. Данные, хранящиеся на SIM-карте, извлекаются в виде байтов. Чтобы получить из этих файлов артефакты, полезные экспертам цифровой криминалистики, необходимо выполнить разбор извлечённых данных. Алгоритм кодирования каждого файла описан в стандарте ISO 7816.

Часть данных представляет собой набор полей со значениями. Для данных подобного вида определённые байты отвечают за значения внутри одного файла SIM-карты. Подобным образом устроено кодирование файла EF_SST, содержащего описание доступных и активированных сервисов SIM-карты. Например, второй байт отображает статус сервиса номеров сокращённого набора, а тринадцатый байт идентифицирует статус сервиса последних набранных номеров.

Для описания более сложных данных используются полноценные алгоритмы кодирования. Например, для декодирования файла

EF_IMSI применяется следующий алгоритм. Первый байт означает длину значащих байтов. Со второго по девятый байт идут значения IMSI. Если оператор сети выбрал значение IMSI меньше 15 символов, остальные полубайты заполняются значением «F». Младший полубайт второго байта (биты 1-4) содержит системные данные, затем начинаются значения IMSI. Каждый байт разделяется на два полубайта, каждый из которых означает одну цифру IMSI. Чтение байтов IMSI происходит справа налево, то есть сначала идёт цифра, полученная из младшего полубайта, затем цифра, полученная из старшего полубайта.

Например, разбор файла EF_IMSI, содержащего значение «08 29 05 02 33 82 65 31 73 65» (через пробел указаны байты), выполнится следующим образом. Первый байт «08» означает длину значащих байтов. То есть закодированный IMSI номер состоит из восьми значащих байтов. Во втором байте «29» младший полубайт содержит системное значение, старший полубайт содержит первую цифру IMSI «2». Третий байт «05» содержит две цифры IMSI, разбор которых выполняется справа налево: сначала идёт цифра «5», полученная из младшего полубайта, затем цифра «0» из старшего. То есть третий байт декодируется в число «50».

Таким образом, результат декодирования IMSI значения из примера равен «250203328561337». По этому идентификатору можно определить, что мобильный код страны тестовой SIM-карты равен «250», код мобильной сети равен «20», а индивидуальный номер мобильного абонента равен «3328561337». Из этих данных следует, что SIM-карта была выпущена в России и принадлежит оператору сети Tele2.

Кодирование одного файла может осуществляться с помощью нескольких различных алгоритмов. Например, для разбора файла контактов EF_ADN существуют четыре стандартизированных алгоритма кодирования данных. Какой вариант алгоритма используется, можно определить по первому байту данных. Выбранный алгоритм кодирования данных зависит от оператора сети, который выпустил SIM-карту. В ходе извлечения данннных с различных SIM-карт было выяснено, что алгоритмы кодирования контактов отличаются на SIM-картах Tele2,

МТС и Мегафон.

4. Архитектура

4.1. Архитектура модуля

Архитектура модуля извлечения данных Сим-карты представлена на рис. 5 (диаграмма компонент UML). Синим цветом обозначены продукты компании «Цифровая корпоративная защита», зелёным цветом обозначен реализованный модуль извлечения данных, оранжевым цветом реализованный модуль разбора извлечённых данных, серым цветом — третьесторонняя библиотека.

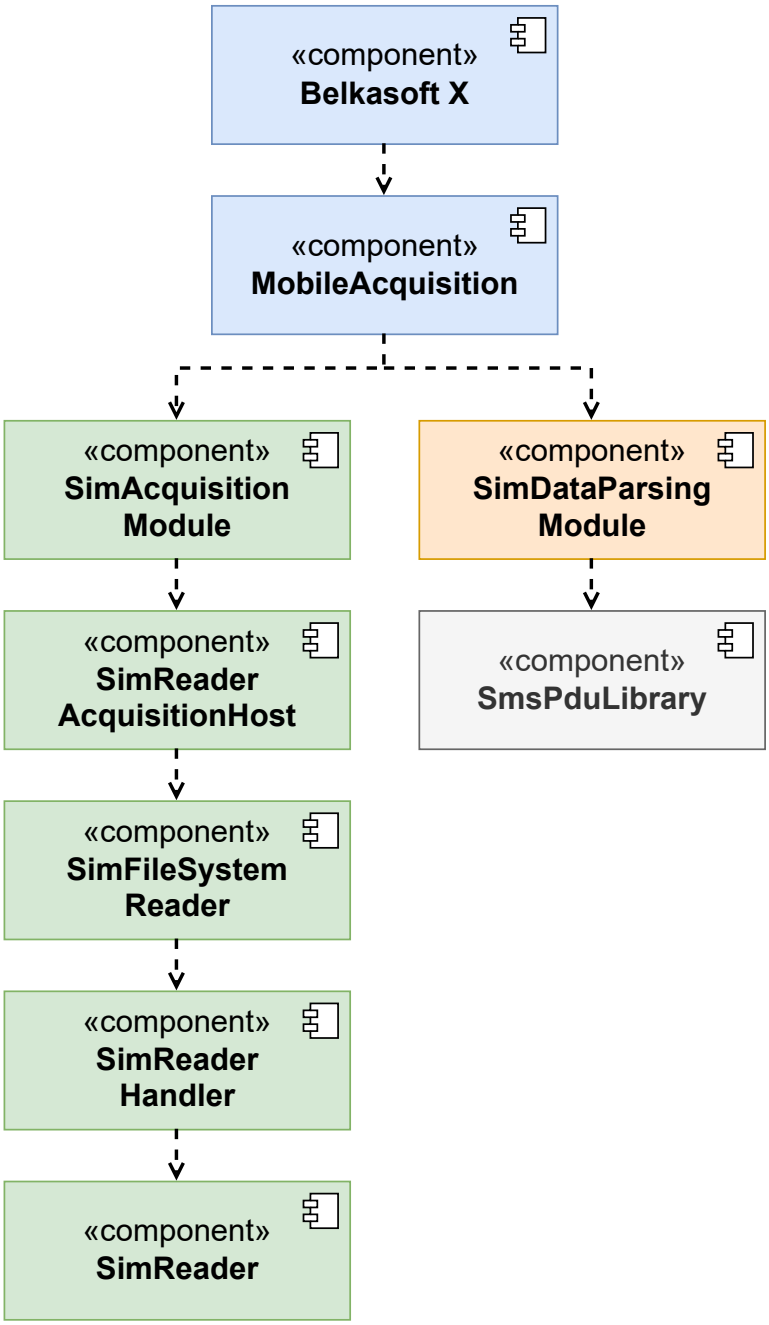

Рис. 5: Диаграмма компонент модуля.

Belkasoft X — инструмент цифровой криминалистики, разработанный для снятия и анализа данных с компьютера, мобильных устройств, облачных хранилищ. Реализован на C# и C++.

MobileAcquisition — модуль, используемый для анализа мобильных устройств и приложений в Belkasoft X. В MobileAcquition есть подмодули, предназначенные для извлечения данных и анализа различных мобильных устройств. Реализован на C#.

SimAcquisitionModule — подмодуль MobileAcquisition, предназначенный для извлечения данных Сим-карты. Модуль интегрирован в продукт Belkasoft X, поэтому реализован на языке программирования C# и C++.

SimReaderAcquisitionHost используется для конфигурации и сохранения извлекаемых с Сим-карты файлов. На считыватель карт отправляются команды, с помощью которых извлекаются данные Сим-карты. Полученные данные сохраняются в бинарные файлы. SimReaderAcquisitionHost реализован на C#.

SimFileSystemReader позволяет извлечь данные всей файловой системы Сим-карты. Последовательно считывается содержимое всех файлов Сим-карты. Считанные данные передаются в SimReaderAcquisitionHost. Реализован на C#.

SimReaderHandler предназначен для взаимодействия с компонентой SimReader. SimReaderHandler является частью модуля извлечения данных Сим-карты SimAcquisitionModule. Реализован на C++/CLI для связи C# и C++ частей модуля.

SimReader предназначен для взаимодействия со считывателем карт. В SimReader реализованы функции выбора и считывания файла Сим-карты. Реализован на C++, поскольку взаимодействие со считывателем карт выполняется с помощью стандартных функций C++.

SimDataParsingModule — подмодуль MobileAcquisition, предназначенный для разборы извлечённых с Сим-карты файлов. Поскольку каждый файл приходится разбирать отдельно, в SimDataParsingModule реализовано несколько классов-разборщиков данных. SMS-сообщения можно перевести в PDU-формат, для разбора таких сообщений используется третьесторонняя библиотека SimPduLibrary. Модуль интегрирован в продукт Belkasoft X, поэтому реализован на языке программирования C#.

SimPduLibrary — третьесторонняя библиотека, предназначенная для декодирования SMS-сообщений из PDU-формата в человекочитаемые значения. SimPduLibrary имеет лицензию GNU Lesser GPL. Эта библиотека была выбрана как легковесное решение для быстрого встра-

ивания в продукт Belkasoft X. Библиотека SimPduLibrary реализована на языке C#.

4.2. Пользовательский интерфейс

Перед извлечением данных Сим-карты пользователю необходимо выбрать COM-порт, к которому подключён считыватель карт. Программно такой порт нельзя определить заранее, потому что считыватели карт имеют разные названия, которые пользователь может изменять.

При выборе модуля извлечения данных Сим-карт в Belkasoft X пользователю демонстрируется окно с выбором COM-порта, представленное на рис. 6. В этом окне отображаются все подключённые по COM-порту устройства.

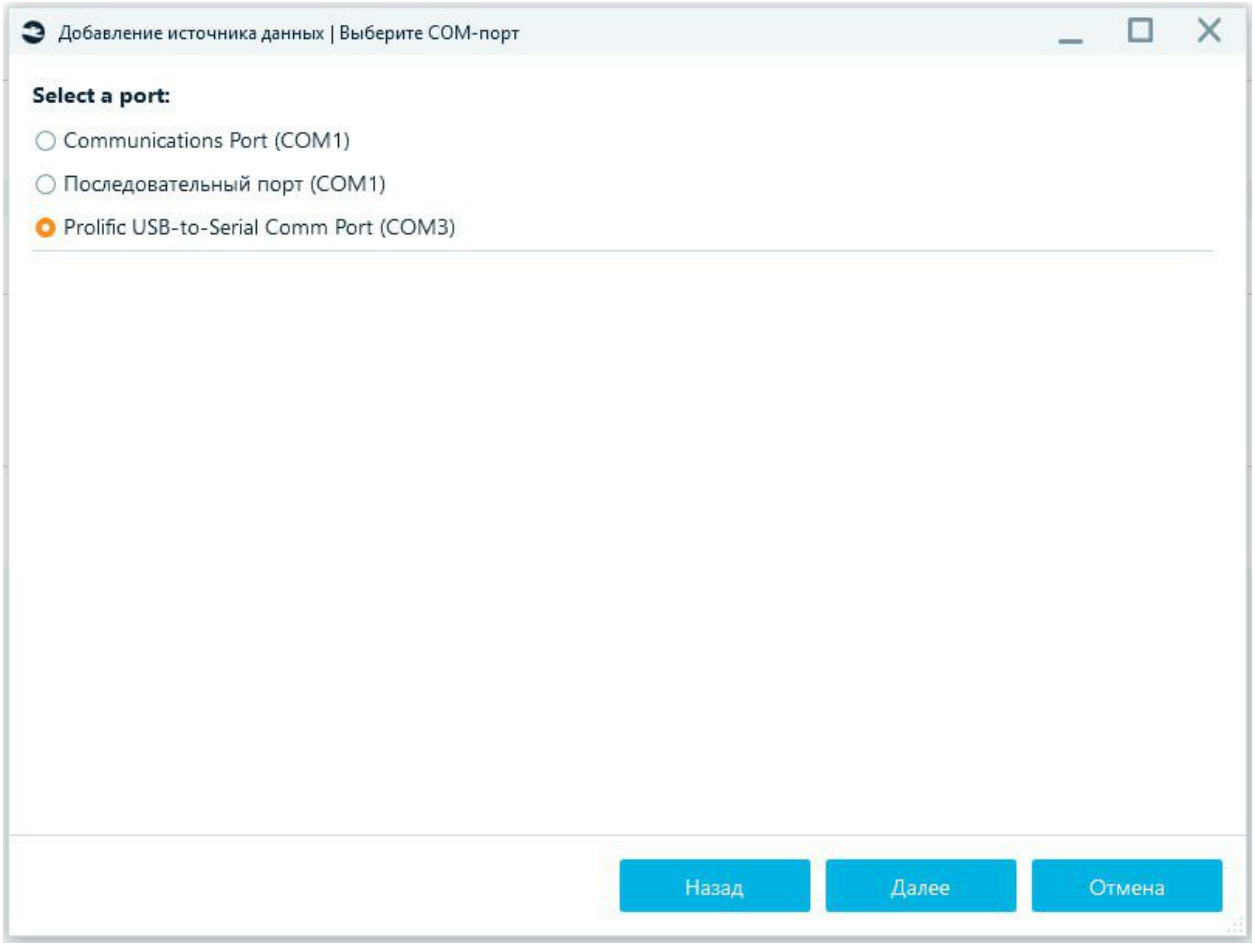

Рис. 6: Окно выбора COM-порта.

Многие считыватели карт в названии содержат значения «USB», «Serial», «Port». Среди всех подключённых по COM-порту устройств выполняется поиск таких значений. По умолчанию на окне выбора COM-порта выбирается устройство, содержащее в названии значения из такого списка. Если устройство не было найдено, выбирается первый элемент списка.

После нажатия на кнопку «Далее» выполняется проверка доступности выбранного COM-порта. Если возникли ошибки при проверке COM-порта, пользователю отображается предупреждение о недоступности выбранного COM-порта (рис. 7). Извлечение данных Сим-карты не начнётся, пока COM-порт не пройдёт проверку.

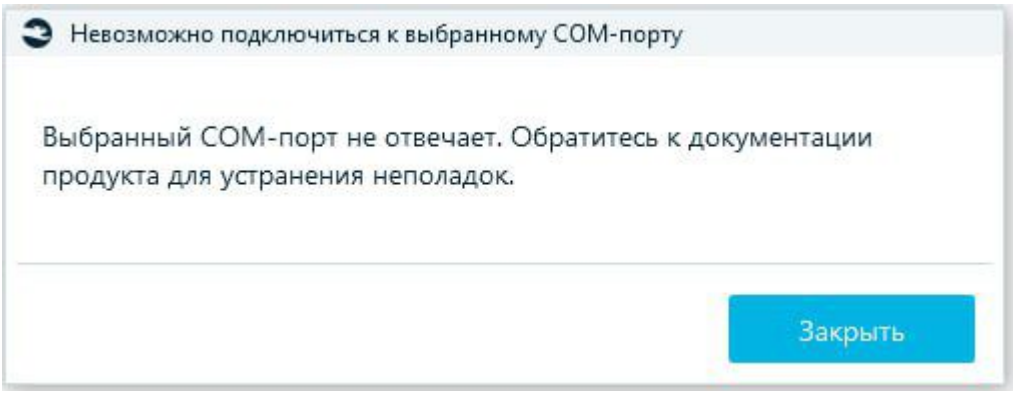

Рис. 7: Сообщение об ошибке проверки COM-порта.

5. Особенности реализации

5.1. Реализация компоненты, взаимодействующей со считывателем карт

Разработанный модуль извлечения данных SIM-карты, представленный на рис. 5 (диаграмма компонент UML), состоит из нескольких частей. Первая часть SimReaderAcquisitionHost отвечает за выбор извлекаемых файлов и сохранение полученных данных в бинарные файлы. Вторая часть SimFileSystemReader отвечает за считывание содержимого файловой системы SIM-карты. Третья часть SimReader отвечает за взаимодействие со считывателем карт. Общение двух частей производится с помощью SimReaderHandler.

Для взаимодействия со считывателем карт необходимо установить обработчик на COM-порт, к которому подключён считыватель карт. Обработчик можно установить с помощью стандартной команды CreateFileA[4]. После открытия обработчика необходимо прочитать ATR-байты[5].

Отправить команду на считыватель карт можно с помощью стандартной функции WriteFile[6], а прочитать ответ с помощью функции ReadFile[7]. Между отправкой команды и получением ответа необходимо подождать некоторое время. Эмпирическим путём было выяснено, что это значение составляет 500 миллисекунд.

Алгоритм извлечения данных SIM-карт представлен на рис. 8 (диаграмма последовательности UML).

[4]https://learn.microsoft.com/en-us/windows/win32/api/fileapi/nf-fileapi-createfilea
[5]https://www.cardlogix.com/glossary/atr-answer-to-reset-smart-card/
[6]https://learn.microsoft.com/en-us/windows/win32/api/fileapi/nf-fileapi-writefile
[7]https://learn.microsoft.com/en-us/windows/win32/api/fileapi/nf-fileapi-readfile

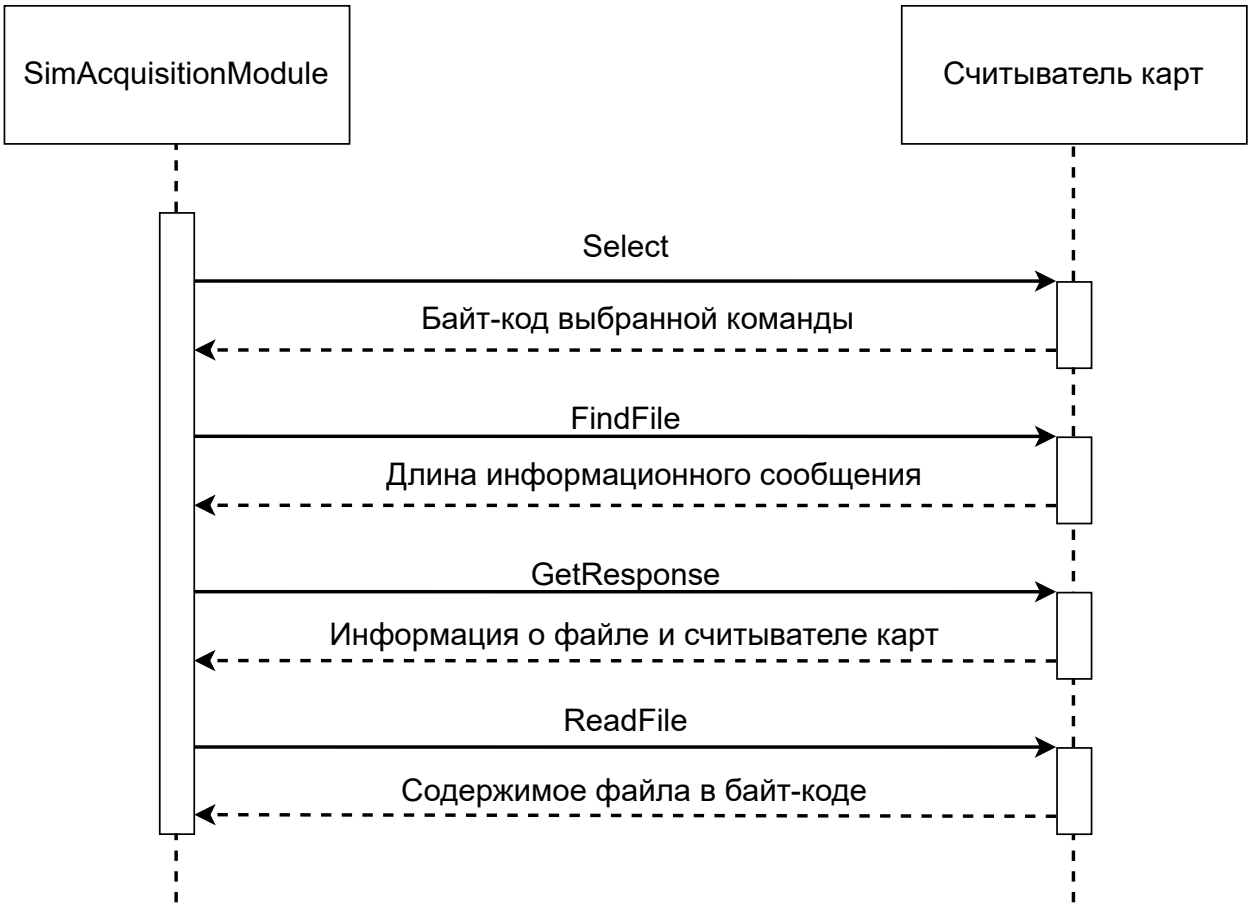

Рис. 8: Алгоритм чтения файла SIM-карты.

Для выбора файла для чтения на SIM-карте необходимо отправить несколько команд на считыватель карт. Первая команда Select предназначена для оповещения считывателя карт о том, что необходимо выбрать файл. В этом запросе указывается специальный код команды, описанный в документации. В ответ считыватель карт повторяет отправленную в байтах команду, добавляя в качестве последнего байта код установленной команды.

Для каждого файла на SIM-карте можно прочитать блок данных, содержащий информацию о считывателе карт: установлен ли PIN-код, число оставшихся попыток ввода PIN-кода, а также другие данные. Также такие блоки содержат длину считываемого файла.

Следующей запрос FindFile направлен на поиск считываемого файла. В запрос передаётся тип (EF или DF) и номер файла. В ответ считыватель карт повторяет отправленные байты, добавляя в качестве последних двух байтов результат исполнения команды и длину блока дан-

ных с информацией о считывателе карт и файле. Если файл не удалось найти, отправляются специальные байты.

Следующий запрос ReadInfoFile направлен на получение блока с данными для считываемого файла. В запросе указывается длина блока, полученная в ответ на предыдущий запрос. В ответ считыватель карт повторяет отправленные байты, добавляя содержимое информационного файла.

Последний запрос ReadFile направлен на считывание файла, который был выбран. В запросе указывается команда считывания файла, а также его длина, которую можно определить из информационного блока. В ответ считыватель карт повторяет отправленные байты, добавляя содержимое файла.

5.2. Реализация компоненты, считывающей файловую систему SIM-карты

Разработанная компонента, считывающая файловую систему SIM-карты, реализована на языке программирования C#. Реализация компоненты представлена на рис. 9 (диаграмма классов UML). Компонента SimFileSystemReader реализована с использованием шаблона проектирования «Шаблонный метод».

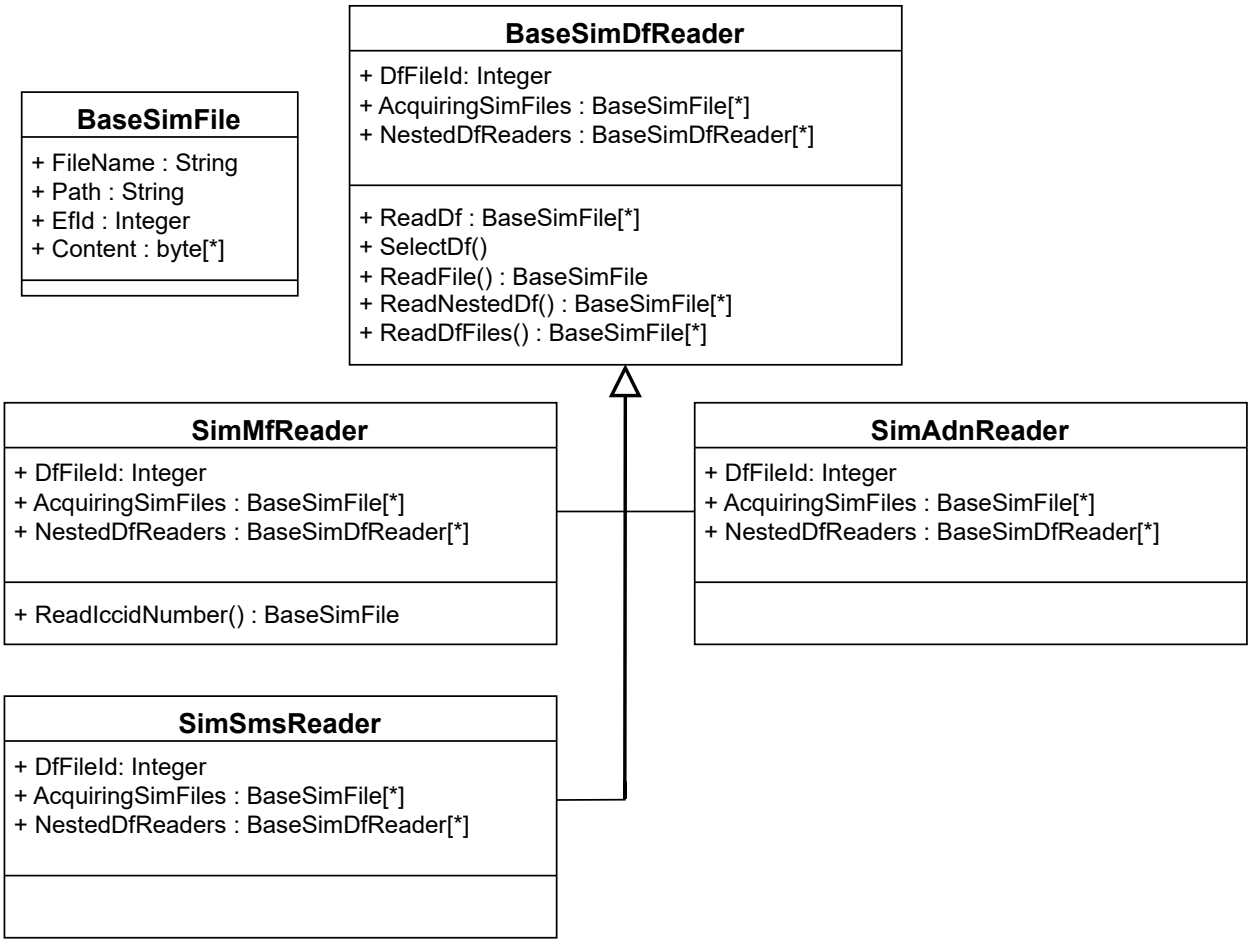

Рис. 9: Устройство компоненты, считывающей файловую систему SIM-карты.

Шаблонный метод ReadDf() и основная функциональность находятся в абстрактном классе BaseSimDfReader. Шаблонный метод ReadDf состоит из нескольких шагов: выбор уровня DF (метод SelectDf), считывание файлов уровня (метод ReadDfFiles), считывание вложенных уровней (метод ReadNestedDf), преобразование считанных данных в базовый класс SIM-файла BaseSimFile (метода ReadFile). Считываемые файлы определяются в поле AcquiringSimFiles. Считыватели вложенных уровней указываются в поле NestedDfReaders. В каждой реализации считывателя файлов определяются файлы и вложенные считыватели.

5.3. Реализация компоненты, разбирающей файловую систему SIM-карты

Разработанная компонента, выполняющая разбор извлечённой файловой системы SIM-карты, реализована на языке программирования C#. Реализация компоненты представлена на рис. 10 (диаграмма классов UML). Компонента SimDataParsingModule реализована с использованием шаблонов проектирования «Шаблонный метод» и «Фабричный метод».

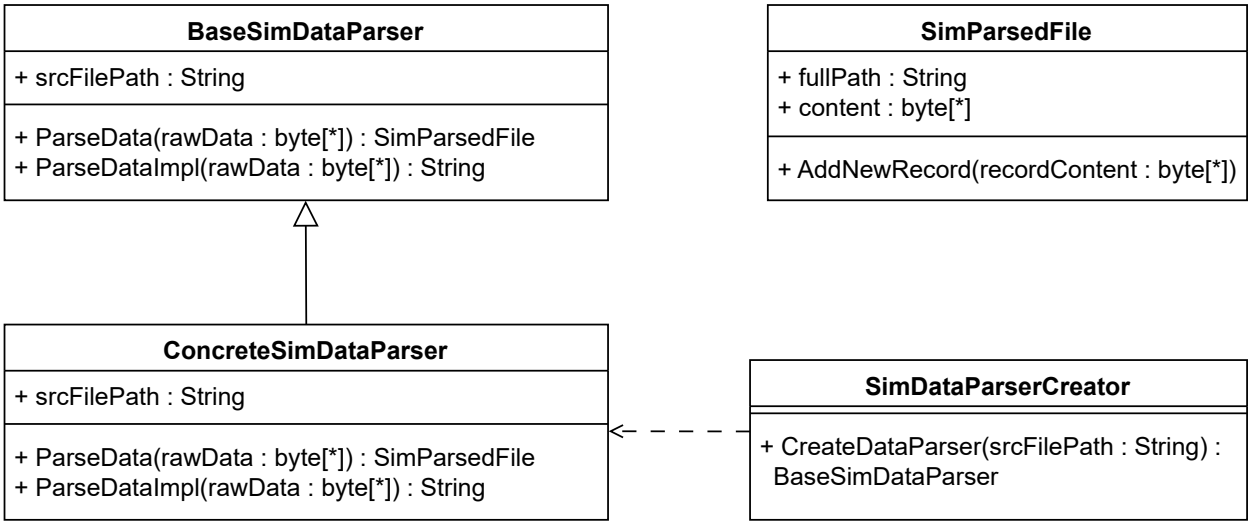

Рис. 10: Устройство компоненты, разбирающей файловую систему SIM-карты.

Шаблонный метод ParseData(rawData : byte[*]) находится в абстрактном классе BaseSimDataParser. В шаблонном методе выполняется безопасный вызов реализации метода разбора данных ParseDataImpl, обработки результата и возврата разобранных данных.

Реализация фабричного метода находится в классе SimDataParserCreator. В методе CreateDataParser(srcFilePath : String) создаётся реализация разборщика конкретного файла SIM-карты.

5.4. Внедрение C++ кода в C#

Компонента, позволяющая извлекать данные SIM-карты, реализована на языке программирования C++. Разработанный модуль извле-

чения данных SIM-карты, а также продукт Belkasoft X, в который производится интеграция, реализованы на языках программирования C# и C++.

Для использования C++ кода в C# проекте была реализована компонента SimReaderHandler. SimReaderHandler реализован на языке программирования C++/CLI, что позволяет вызывать код, написанный на C++, в C#-проекте. В качестве технологий для внедрения C++ кода в C#-проект рассматривались технологии C++/CLI и P/Invoke. Технология C++/CLI была выбрана для упрощения отладки C++-кода из C#-проекта.

Компонента SimReaderHandler реализована с использованием шаблона проектирования «Фасад». Доступны только две функции: выбрать файл, считать содержимое выбранного файла. Вся остальная логика скрыта внутри компоненты SimReader.

6. Тестирование и апробация

Апробация реализованного модуля извлечения данных Сим-карты проводилась с использованием тестового считывателя карт. Были извлечены данные с пяти Сим-карт различных операторов: две Сим-карты Tele2, одна Сим-карта Megafon, одна Сим-карта Beeline и одна Сим-карта MTS. Со всех Сим-карт удалось извлечь файловую систему Сим-карты и выполнить разбор извлечённых данных. Среди извлечённых данных были найдены следующие артефакты: номер IMSI, телефонная книгу номеров сокращённого набора ADN, отправленные и полученные сообщения SMS.

Реализованная функциональность прошла проверку кода, была интегрирована в исходный код проекта Belkasoft X и была проверена командой тестирования компании «Цифровая Корпоративная Защита».

7. Заключение

В ходе данной работы были получены следующие результаты.

- Проведён обзор существующих аналогов: E3: Electronic Evidence Center, Oxygen Forensics Detective, СимLab, Osmo-sim-auth, DualSimCard.
- Выяснен принцип извлечения данных Сим-карты: команды и ответы на них отправляются в байтах согласно стандарту ISO 7816.
- Спроектирован и реализован модуль, извлекающий и выполняющий разбор файловой системы Сим-карты (C++, C#, C++/CLI).
- Выполнена интеграция разработанного модуля в Belkasoft X. Реализованная функциональность была добавлена в исходный код проекта.

Данную работу планируется продолжать в следующем семестре и довести до уровня ВКР. В рамках ВКР поставлены задачи:

- Реализовать окно верификации PIN-кода и PUK-кода Сим-карты.
- Провести тестирование и апробацию разработанного модуля.

Код проекта закрыт и принадлежит компании ООО «Цифровая корпоративная защита».

Список литературы

[1] Charles Griffiths. The Latest 2022 Cyber Crime Statistics. — AAG, 2022. URL: https://aag-it.com/the-latest-2022-cyber-crime-statistics/ (дата обращения: 23.12.23).

[2] How to Handle Data Acquisition in Digital Forensics. — EC-Council Cybersecurity Exchange, 2022. — URL: https://www.eccouncil.org/cybersecurity-exchange/computer-forensics/data-acquisition-digital-forensics/ (дата обращения: 23.12.22).

[3] Manuel Rozewski. What Information Is Stored On A SIM Card? — SimOptions, 2021. URL: https://www.simoptions.com/sim-card-information/ (дата обращения: 23.12.23).

[4] Milan. What is SIM Card Reader and How Does it Work? — Hybrid Sim, 2021. URL: https://hybridsim.com/sim-card-reader/ (дата обращения: 23.12.23).

[5] Warlock. SIM card forensics: An introduction. — Infosec, 2013. — URL: https://resources.infosecinstitute.com/topic/sim-card-forensics-introduction/ (дата обращения 23.12.22).

[6] Linux File System. Javatpoint. — URL: https://www.javatpoint.com/linux-file-system (дата обращения: 23.12.23).

[7] Content for TS 31.102. Tech-invite. — URL: https://www.tech-invite.com/3m31/toc/tinv-3gpp-31-102_c.html#e-4-2-2 (дата обращения: 23.12.23).

[8] Content for TS 31.102. Tech-invite. — URL: https://www.tech-invite.com/3m31/toc/tinv-3gpp-31-102_r.html#e-4-4-2-3 (дата обращения: 23.12.23).

[9] E3: Electronic Evidence Examine. — Paraben, 2022. — URL: https://paraben.com/ (дата обращения: 23.12.23).

[10] Oxygen Forensics Detective. — Oxygen Forensics, 2022. — URL: https://www.oxygen-forensic.com/en/products/oxygen-forensic-detective (дата обращения: 23.12.23).

[11] SimLAB. — Kamil Wartanowicz, 2016. — URL: https://github.com/kamwar/simLAB (дата обращения: 23.12.23).

[12] Osmo-sim-auth. — Gerard Pinto, 2017. — URL: https://github.com/GerardPinto/osmo-sim-auth (дата обращения: 23.12.23).

[13] DualSIMCard. — Piotr Zerynger, 2019. — URL: https://github.com/ITger/DualSIMCard (дата обращения: 23.12.23).

[14] Free Serial Port Monitor — COM Port Monitoring, 2022 — URL: https://www.com-port-monitoring.com/#free-monitor (дата обращения: 23.12.23).

[15] Smart Card Standards. — QCard. — URL: https://www.q-card.com/about-us/smart-card-standards/page.aspx?id=1461 (дата обращения: 23.12.23).