Санкт-Петербургский государственный университет

Степырев Даниил Федорович

Производственная практика

Извлечение данных SIM-карты с использованием считывателя карт

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

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

Санкт-Петербург
2024

Оглавление

1. Введение **3**

2. Постановка задачи **5**

3. Обзор **6**

3.1. Файловая система SIM-карты 6

3.2. Считыватель карт 8

3.3. Способы извлечения данных SIM-карты 8

3.4. Обзор аналогов 9

3.5. Работа с SIM-картой через COM-порт 12

3.6. Извлечение данных SIM-карты 14

3.7. Подтверждение PIN-кода SIM-карты 14

3.8. Разбор извлечённых данных SIM-карты 16

4. Архитектура **19**

4.1. Архитектура модуля 19

4.2. Пользовательский интерфейс 21

5. Особенности реализации **25**

5.1. Реализация компоненты, взаимодействующей со считывателем карт 25

5.2. Реализация компоненты, считывающей файловую систему SIM-карты 27

5.3. Реализация компоненты, разбирающей файловую систему SIM-карты 29

5.4. Внедрение C++ кода в C# 29

6. Тестирование и апробация **31**

7. Заключение **32**

Список литературы **33**

1. Введение

С развитием технологий в современном мире возрастает и число цифровых преступлений [3]. Это могут быть кражи личных данных, распространение незаконной информации, вмешательство в работу приложений и другие виды преступлений.

Цифровая криминалистика является наукой, которая помогает обнаруживать, фиксировать и исследовать компьютерные доказательства для подтверждения противоправных действий [19]. Цифровая криминалистика используется в судебной практике и не предназначена для злонамеренных действий.

Для обнаружения и анализа доказательств преступлений эксперты цифровой криминалистики снимают данные с различных устройств [9]. Эти данные могут быть извлечены из памяти цифровых устройств, внешних запоминающих устройств или облачных хранилищ данных. Одним из таких устройств является SIM-карта.

На SIM-карте хранится идентификационная информация о пользователе: международный номер мобильного абонента (IMSI), ключ аутентификации пользователя (KI). Кроме того, на SIM-карте также могут храниться данные о пользователе, такие как телефонная книга, журнал звонков и SMS-сообщения [16]. Эта информация полезна для экспертов в области цифровой криминалистики.

Чтение SIM-карты может осуществляться с помощью специальных устройств, называемых считывателями карт [13]. SIM-карта вставляется в соответствующий разъем устройства, которое затем подключается к компьютеру.

Интерес к снятию данных с SIM-карт возник у компании «Цифровая корпоративная защита» при разработке продукта Belkasoft X [2]. Belkasoft X — инструмент цифровой криминалистики, разработанный для снятия и анализа данных с компьютера, мобильных устройств, облачных хранилищ.

На момент написания данной работы существует несколько работающих проектов, поддерживающих извлечение данных SIM-карты с

использованием считывателя карт. Однако эти проекты либо не обеспечивают полного снятия и анализа файловой системы SIM-карты, либо представляют собой условно-бесплатные ограниченные версии.

Тем не менее с помощью инструментов обратной разработки можно изучить принципы снятия данных с SIM-карты и создать модуль, позволяющий извлечь всю файловую систему SIM-карты с использованием считывателя карт. Реализация подобной функциональности для коммерческого продукта Belkasoft X является целью данной работы.

2. Постановка задачи

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

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

3. Обзор

3.1. Файловая система SIM-карты

Файловая система SIM-карты организована по принципу древовидной структуры, которая часто используется для упорядочивания файлов и каталогов, подобно тому, как это реализовано в файловой системе Linux [11]. Однако в файловой системе SIM-карты каталог также представляется в виде файла.

Наивысшим уровнем файловой системы SIM-карты является основной файл MF (Master File), содержащий все остальные файлы, хранящиеся на SIM-карте [4]. Кроме него, существуют два вида файлов: элементарные и вложенные.

Элементарные файлы EF (Elementary File) содержат только данные и не могут включать в себя другие файлы. Данные в них хранятся в байтовом виде. Примером такого файла является EF_IMSI, содержащий информацию об IMSI-номере SIM-карты [20].

Вложенные файлы DF (Dedicated File) могут содержать в себе другие файлы, включая как элементарные, так и другие вложенные. Часто данные организованы в несколько элементарных файлов, хранящихся внутри одного вложенного файла. Примером таких файлов может быть телефонная книга ADN сокращённого набора [8].

Структура файловой системы SIM-карты показана на рис. 1 в виде UML-диаграммы компонентов. На верхнем уровне файловой системы SIM-карты располагается только корневой файл MF, содержащий в себе вложенные файлы с описанием уровня GSM (файл DF_GSM) и уровня TELECOM (файл DF_TELECOM). Кроме вложенных файлов на уровне MF имеется элементарный файл, описывающий уникальный серийный номер ICCID SIM-карты (файл EF_ICCID) [12].

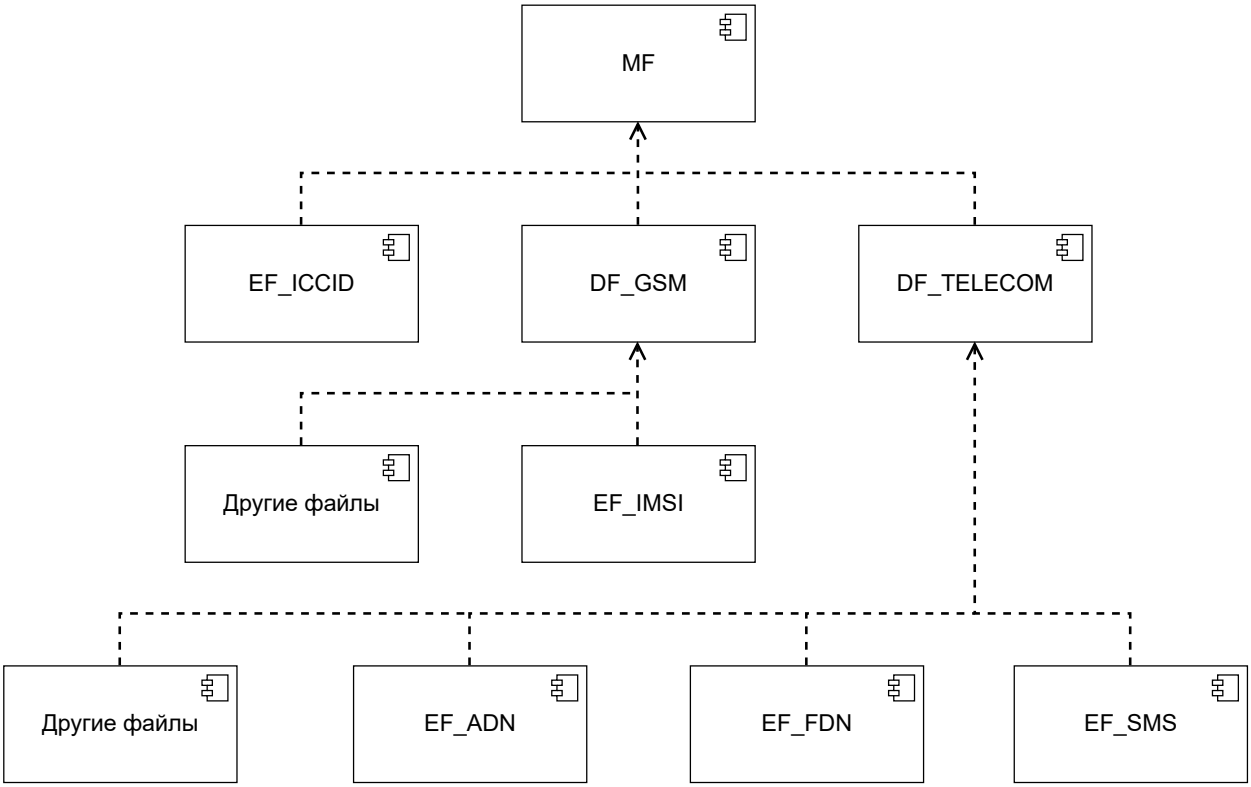

Рис. 1: Файловая система SIM-карты.

На уровне GSM располагаются файлы, содержащие информацию, связанную с мобильной сетью. На этом уровне также находится файл EF_IMSI, хранящий международный идентификатор мобильного абонента.

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

SIM-карту можно защитить, установив PIN[1]-код, действующий аналогично паролю. Для доступа к данным на SIM-карте пользователю требуется ввод PIN-кода. После трёх неудачных попыток ввода PIN-кода SIM-карта блокируется и требует ввода PUK[2]-кода. Без ввода PUK-кода SIM-карта остаётся заблокированной. После десяти неверных попыток ввода PUK-кода SIM-карта полностью блокируется.

[1]PIN — Personal Identification Number.

[2]PUK — Personal Unlocking Key.

3.2. Считыватель карт

Считыватель карт — специальное устройство, предназначенное для взаимодействия с SIM-картой. Оно обычно имеет отдельное отверстие для подключения SIM-карты и подключается к компьютеру через USB-порт. Считыватель карт позволяет читать данные SIM-карты через COM-порт. Пример считывателя карт, используемого в данной работе, представлен рис. 2.

Рис. 2: Изображение считывателя карт.

3.3. Способы извлечения данных SIM-карты

Существует несколько способов извлечения данных с SIM-карты. Первый способ — это физическое извлечение, которое предполагает прямой доступ к физическому носителю данных на SIM-карте. Для этого используются специализированные считыватели карт, которые позволяют подключить SIM-карту к компьютеру для последующего анализа. Такой метод обеспечивает полный доступ к физическим данным на SIM-карте, включая удалённые и скрытые файлы.

Другой метод — логическое извлечение, при котором данные извлекаются через программное обеспечение, взаимодействующее с устройством, содержащим SIM-карту. В этом случае данные извлекаются с использованием стандартных протоколов связи и не требуют физиче-

ского доступа к SIM-карте. Однако этот метод ограничен доступом к некоторым данным, и могут возникать проблемы совместимости с различными устройствами и протоколами.

Поскольку считыватель карт предполагает прямое физическое чтение данных с SIM-карты, он считается более надёжным и точным методом извлечения. Процесс извлечения в этом случае не зависит от производителя и модели устройства, так как взаимодействие происходит непосредственно с SIM-картой. По этой причине для извлечения данных SIM-карты было решено использовать физический метод с использованием считывателя карт.

3.4. 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 использованием ключевых слов «SIM card», «Acquisition», «Reader», «tools».

3.4.1. E3: Electronic Evidence Examine

E3: Electronic Evidence Examine — продукт, представленный компанией Paraben, позволяющий извлекать данные SIM-карты [5]. Инструмент был ранее известен как SIMCon, но позже был интегрирован как модуль в продукт E3: Electronic Evidence Examiner. Этот продукт позволяет не только извлекать и анализировать всю файловую систему SIM-карты, но и проводить верификацию PIN-кода. Хотя продукт платный (стоимость лицензии составляет 1895$ в год), доступна бесплатная версия на 30 дней.

3.4.2. Oxygen Forensics Detective

Oxygen Forensics Detective — продукт компании Oxygen Forensics, позволяющий извлекать данные SIM-карты [14]. Он поддерживает извлечение и анализ файловой системы SIM-карты, а также верификацию

PIN-кода. Этот продукт также является платным (стоимость лицензии составляет 8090€ в год), но есть бесплатная версия на 20 дней.

3.4.3. SimLAB

SimLAB — продукт с открытым исходным кодом, который позволяет извлекать файловую систему SIM-карты [10]. Извлечение данных происходит в бинарном формате, и инструмент не позволяет разбирать извлечённые файлы. Проект не имеет активной поддержки, а последнее обновление было в 2016 году.

3.4.4. Osmo-sim-auth

Osmo-sim-auth — продукт с открытым исходным кодом, позволяющий извлекать файловую систему SIM-карты [7]. Инструмент помимо извлечения файловой системы SIM-карты также поддерживает верификацию PIN-кода. Однако инструмент не предоставляет возможности анализа извлечённых файлов. Последнее обновление проекта было в 2017 году.

3.4.5. DualSIMCard

DualSIMCard — продукт с открытым исходным кодом, предназначенный для извлечения данных SIM-карты [15]. Однако инструмент способен извлекать только часть файлов SIM-карты и не обеспечивает доступ к файлам, содержащим пользовательские данные. DualSIMCard также не поддерживает разбор извлечённых данных и проверку PIN-кода. Последнее обновление проекта было в 2019 году.

3.4.6. Сравнение аналогов

Таблица 1 представляет обзор рассмотренных аналогов. Большинство из них способны извлекать файловую систему SIM-карты, но не все обеспечивают полный разбор данных. Только платные продукты, такие

как E3: Electronic Evidence Examine и Oxygen Forensics Detective, предоставляют такую возможность. Продукты simLAB и DualSIMCard позволяют лишь извлекать данные SIM-карты, но не разбирать их структуру. Используя эти инструменты, нельзя гарантированно извлечь и проанализировать 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 и Oxygen Forensics Detective [5, 14] всё ещё поддерживаются. Последние обновления в про-

ектах SimLAB, Osmo-sim-auth и DualSIMCard [10, 15] датированы 2016, 2017 и 2019 годами соответственно.

Большинство из рассмотренных продуктов включает в себя функцию верификации PIN-кода SIM-карты, за исключением проекта DualSIMCard. Однако с использованием инструментов с открытым исходным кодом нельзя гарантированно извлечь и разобрать всю файловую систему SIM-карты. Продукты SimLAB, Osmo-sim-auth и DualSIMCard [7, 10, 15] не смогли извлечь данные с SIM-карты, используя тестовый считыватель карт.

3.5. Работа с SIM-картой через COM-порт

Поскольку с помощью инструментов с открытым исходным кодом не удалось выполнить извлечение данных SIM-карты, для проверки работоспособности считывателя SIM-карты и проверки идей относительно принципов извлечения данных были применены средства обратной разработки. Процесс включал перехват трафика между считывателем карт и одним из доступных продуктов. Для этого использовался бесплатная версия продукта Serial Port Monitor [17]. Архитектура решения представлена на рис. 3 (диаграмма последовательности UML).

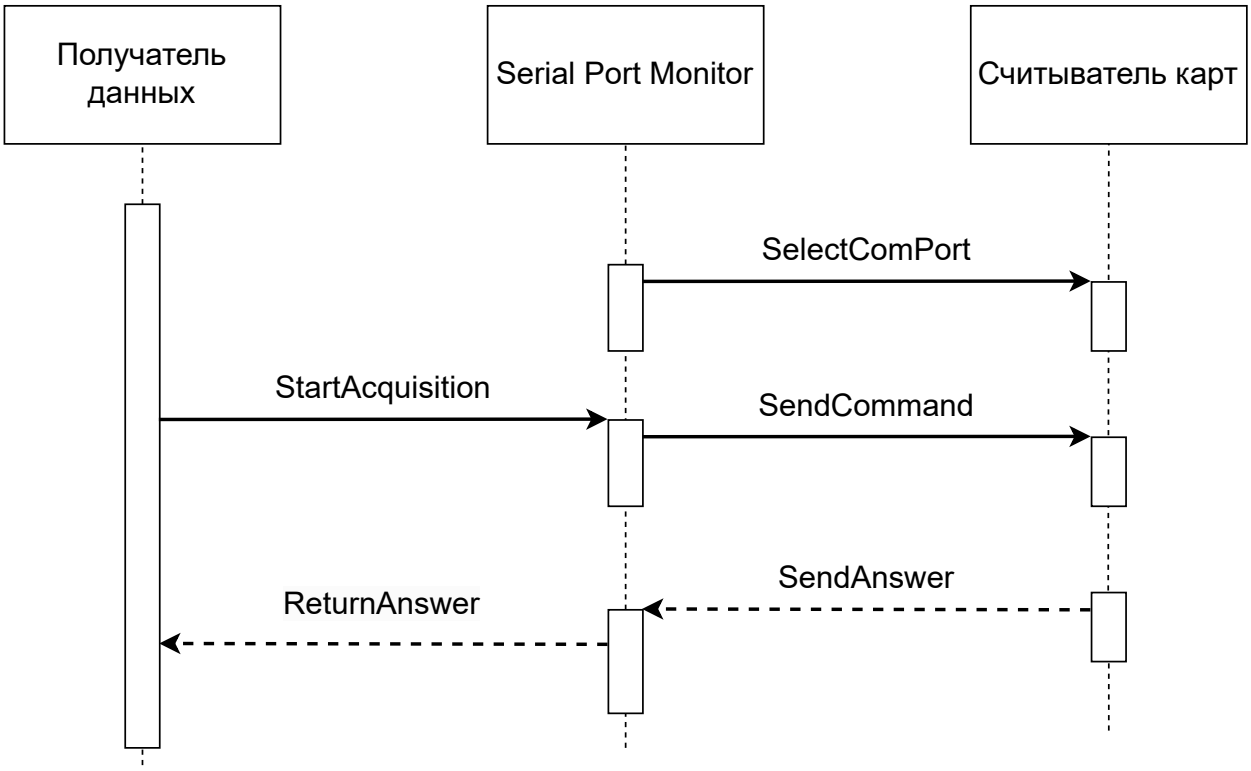

Рис. 3: Принципиальная схема перехвата команд.

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 и получить ответы 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).

3.6. Извлечение данных SIM-карты

В результате анализа удалось определить, что взаимодействие со считывателем карт выполняется в соответствии со стандартом смарт-карт ISO 7816 [18].

Для доступа к файлам SIM-карты необходимо перейти в соответствующую директорию, выбрав нужный файл. Это делается с помощью команды выбора файла. Для доступа к вложенным директориям сначала необходимо перейти в родительскую директорию.

Извлечение данных с SIM-карты может быть ограничено установленным PIN-кодом. Для проверки наличия кода на SIM-карте отправляется специальный запрос. Полученный ответ содержит информацию о наличии PIN-кода и оставшихся попытках его ввода.

При вводе PIN-кода пользователю предоставляется три попытки. После трёх неверных попыток SIM-карта переходит в режим ввода PUK-кода. После десяти неудачных попыток ввода PUK-кода SIM-карта блокируется.

После считывания файла необходимо выполнить разбор полученного ответа от считывателя карт. Сначала необходимо удалить из ответа информацию об исполненной команде. Затем оставшиеся байты следует разобрать согласно алгоритму кодирования, указанному в стандарте ISO 7816.

3.7. 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.

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.

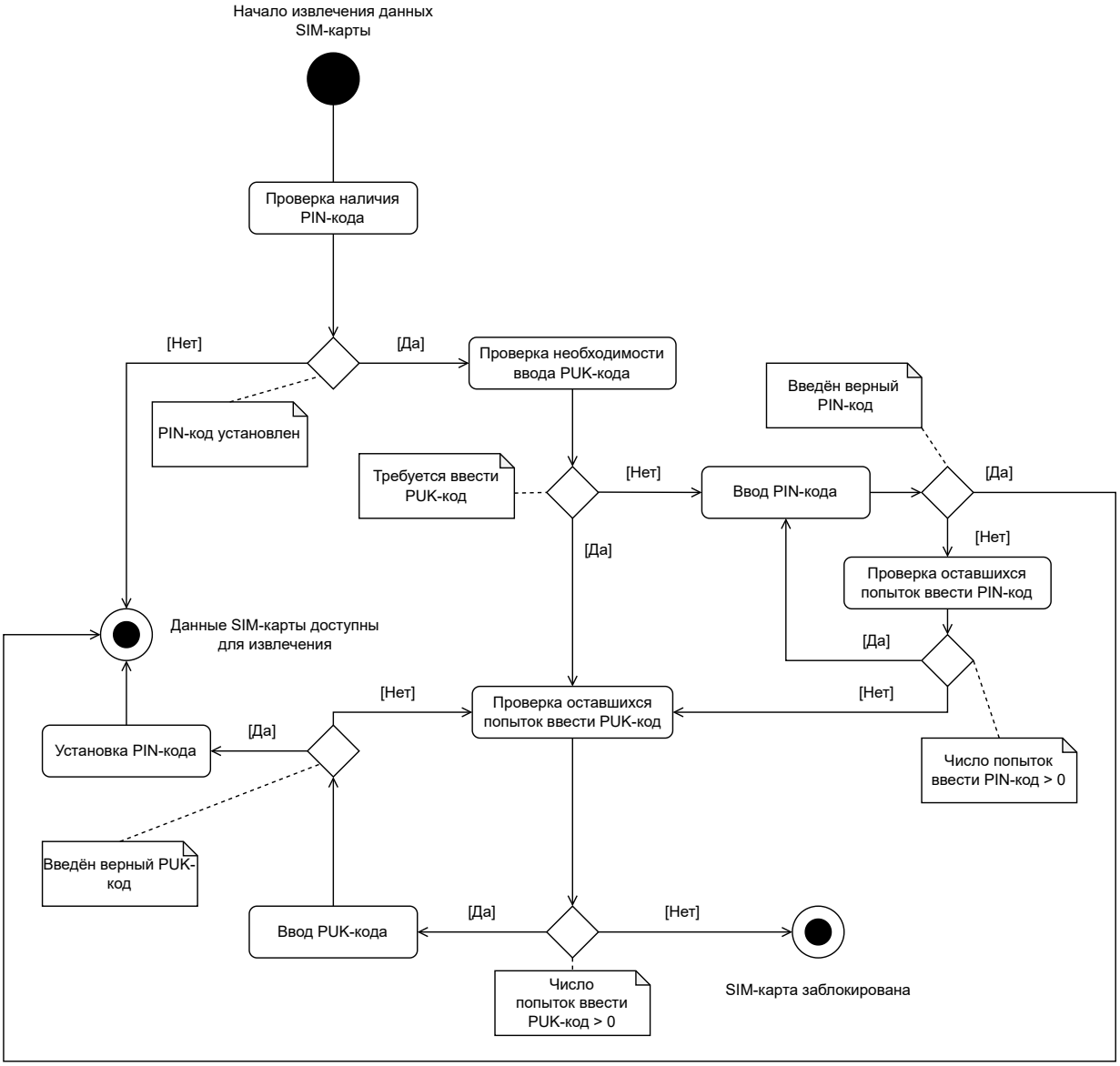

Рис. 4: Алгоритм подтверждения PIN-кода.

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.

3.8. Разбор извлечённых данных SIM-карты

После извлечения данных SIM-карты формируется набор файлов, структура которых повторяет файловую систему SIM-карты. Данные, хранящиеся на SIM-карте, представлены в виде байтов. Для получения артефактов, важных для экспертов цифровой криминалистики, необходимо проанализировать извлеченные данные. Алгоритм кодирования каждого файла описан в стандарте ISO 7816.

Часть данных представляет собой набор полей со значениями. Для данных такого типа определённые байты отвечают за значения внутри одного файла SIM-карты. Аналогичным образом устроено кодирование файла EF_SST, содержащего описание доступных и активированных сервисов SIM-карты [6]. Например, второй байт отображает статус сервиса номеров сокращённого набора, а тринадцатый байт идентифици-

рует 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 значимых байтов. Со второго по девятый байт идут значения IMSI, при этом младший полубайт второго байта (биты 1-4) содержит только системные данные. Остальные байты разбивается на два полубайта, где каждый означает одну цифру IMSI. Чтение байтов IMSI происходит справа налево, то есть сначала идет цифра, полученная из младшего полубайта, затем цифра, полученная из старшего полубайта. Если оператор сети выбрал значение IMSI меньше 15 символов, остальные полубайты заполняются значением «F».

Например, разбор файла EF_IMSI, содержащего значение «08 29 05 02 33 82 65 31 73 65» (через пробел указаны байты в шестнадцатеричной системе счисления), выполняется следующим образом. Первый байт «08» означает длину значимых байтов, следовательно, закодированный IMSI состоит из восьми значимых байтов. Во втором байте «29» младший полубайт содержит системное значение, а старший полубайт содержит первую цифру IMSI «2». Третий байт «05» содержит две цифры IMSI, разбор которых выполняется справа налево: сначала идет 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 несколькими различными алгоритмами. 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 данных 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 SIM-карты представлена на рис. 5 (диаграмма компонентов 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.

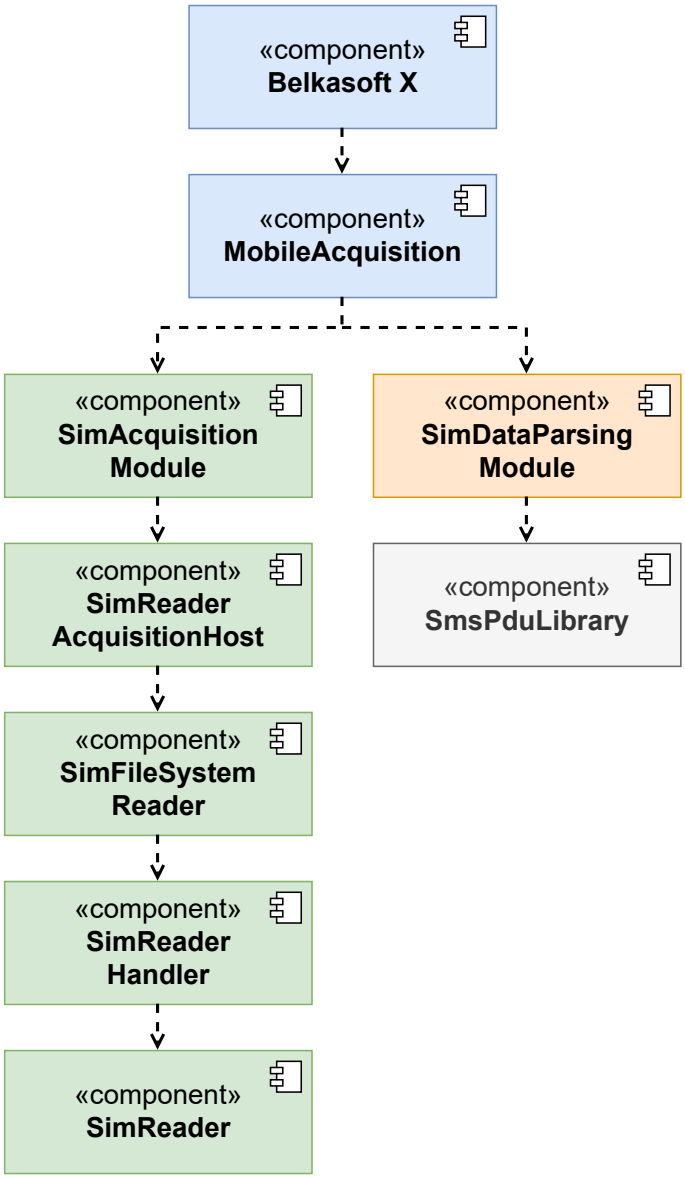

Рис. 5: Диаграмма компонентов модуля.

Belkasoft X — это инструмент цифровой криминалистики, специализирующийся на снятии и анализе данных с компьютеров, мобильных устройств и облачных хранилищ. Реализован на C# и C++.

MobileAcquisition — это модуль, используемый для анализа мобильных устройств и приложений в Belkasoft X. В состав MobileAcquisition входят подмодули, предназначенные для извлечения данных и анализа различных мобильных устройств. Он также реализован на языке программирования C#.

SimAcquisitionModule — это подмодуль MobileAcquisition, разработанный для извлечения данных с SIM-карт. Этот модуль интегрирован в продукт Belkasoft X и поэтому реализован на языке программирования C# и C++.

SimReaderAcquisitionHost используется для конфигурации и сохранения извлечённых файлов с SIM-карт. На считыватель карт отправляются команды для извлечения данных с SIM-карт, после чего полученные данные сохраняются в бинарные файлы. SimReaderAcquisitionHost реализован на языке программирования C#.

SimFileSystemReader позволяет извлечь данные из файловой системы SIM-карт. Он последовательно считывает содержимое всех файлов SIM-карт и передает считанные данные в SimReaderAcquisitionHost. Этот модуль также реализован на языке программирования C#.

SimReaderHandler предназначен для взаимодействия с компонентой SimReader. Он является частью модуля извлечения данных с SIM-карт SimAcquisitionModule и реализован на языке программирования C++/CLI для связи C# и C++ частей модуля.

SimReader предназначен для взаимодействия со считывателем карт. Он реализован на языке программирования C++ и содержит функции выбора и считывания файлов с SIM-карт, поскольку взаимодействие со считывателем карт осуществляется с использованием функций C++.

SimDataParsingModule — это подмодуль MobileAcquisition, предназначенный для разбора извлечённых файлов с SIM-карт. Поскольку каждый файл требуется разбирать отдельно, в SimDataParsingModule реализовано несколько классов-разборщиков данных. Например, SMS-

сообщения могут быть переведены в PDU-формат, и для их разбора используется третьесторонняя библиотека SimPduLibrary. Этот модуль также интегрирован в продукт Belkasoft X и реализован на языке программирования C#.

SimPduLibrary — третьесторонняя библиотека, предназначенная для декодирования SMS-сообщений из PDU-формата в человекочитаемые значения. Она имеет лицензию GNU Lesser GPL и была выбрана как легковесное решение для быстрого встраивания в продукт Belkasoft X. Библиотека SimPduLibrary также реализована на языке программирования C#.

4.2. Пользовательский интерфейс

Перед извлечением данных SIM-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 SIM-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.

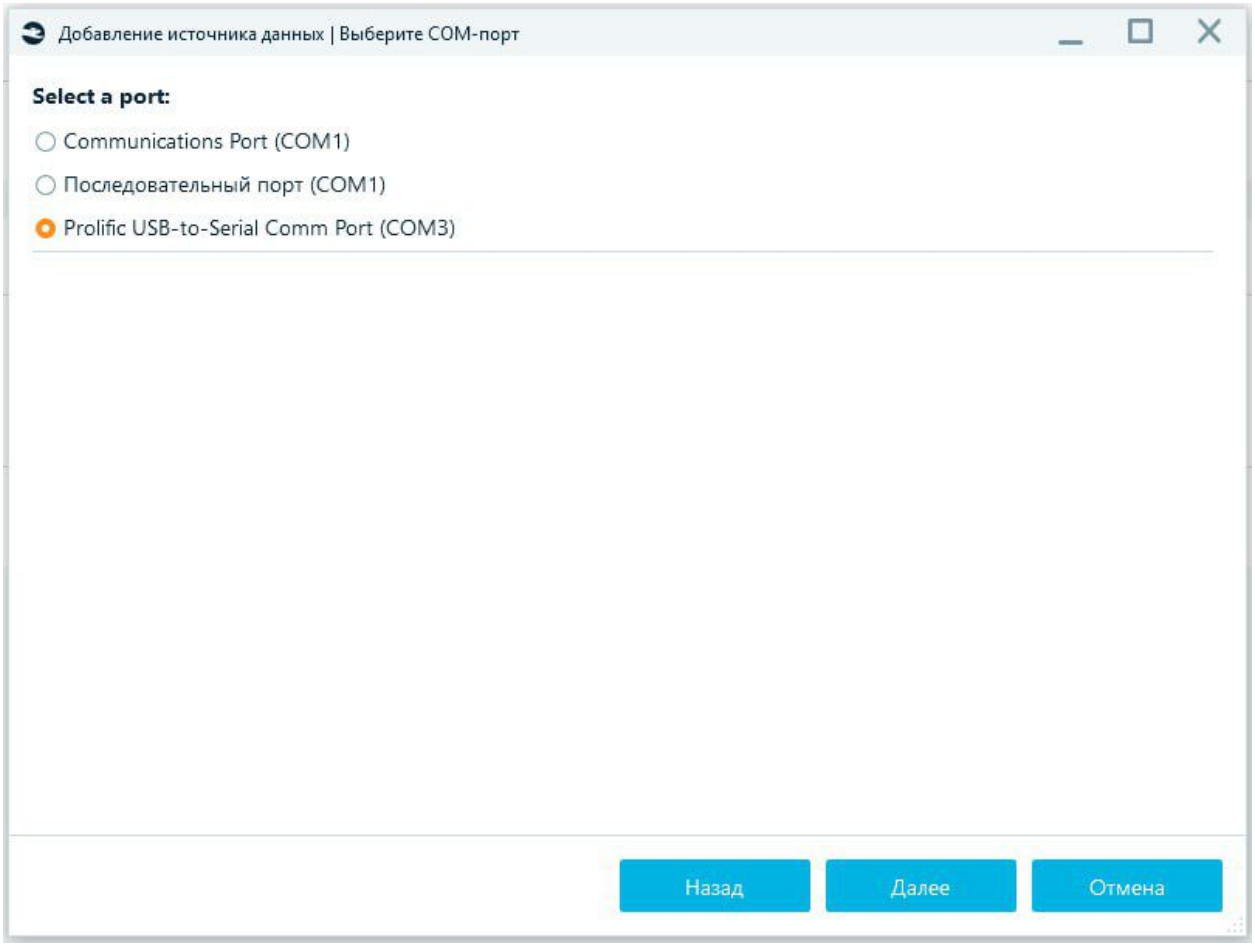

Рис. 6: Окно выбора COM-порта.

Многие считыватели карт обычно имеют в названии ключевые слова, такие как «USB», «Serial» или «Port». При подключении к компьютеру производится поиск устройств, которые используются через COM-порт, и проверяется наличие указанных ключевых слов в их названиях. По умолчанию модуль выбирает первое найденное устройство из этого списка для использования. В случае, если устройство не обнаружено, программа автоматически выбирает первое устройство из списка COM-портов.

После того как пользователь нажимает кнопку «Далее», программа проверяет доступность выбранного COM-порта. Если в процессе проверки возникают ошибки, то пользователю выводится предупреждение, что выбранный COM-порт недоступен (рис. 7). Извлечение данных с SIM-карты не будет начато до тех пор, пока проверка COM-порта не будет успешно завершена.

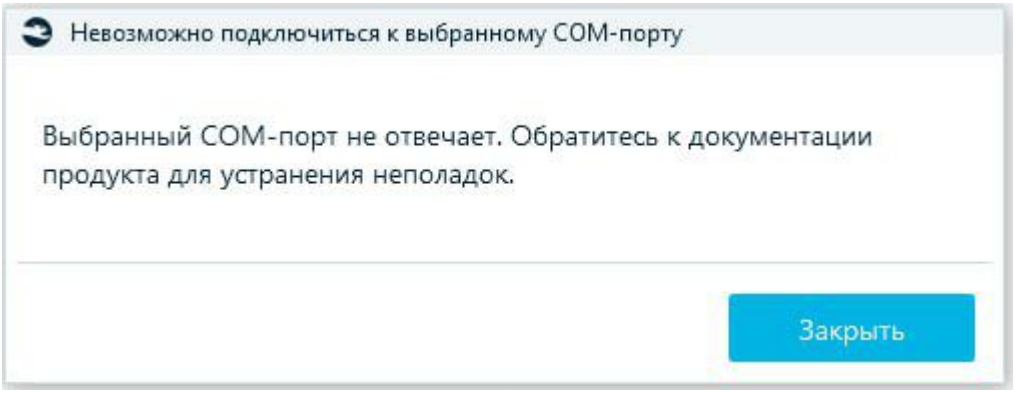

Рис. 7: Сообщение об ошибке проверки COM-порта.

После проверки COM-порта производится проверка наличия установленного PIN или PUK-кода. Если такой код установлен, открывается окно для ввода PIN-кода (рис. 8). На этом окне отображается тип необходимого кода (PIN или PUK), а также предоставляется поле для ввода значения. Переход на следующую страницу невозможен до тех пор, пока не будет введен правильный код.

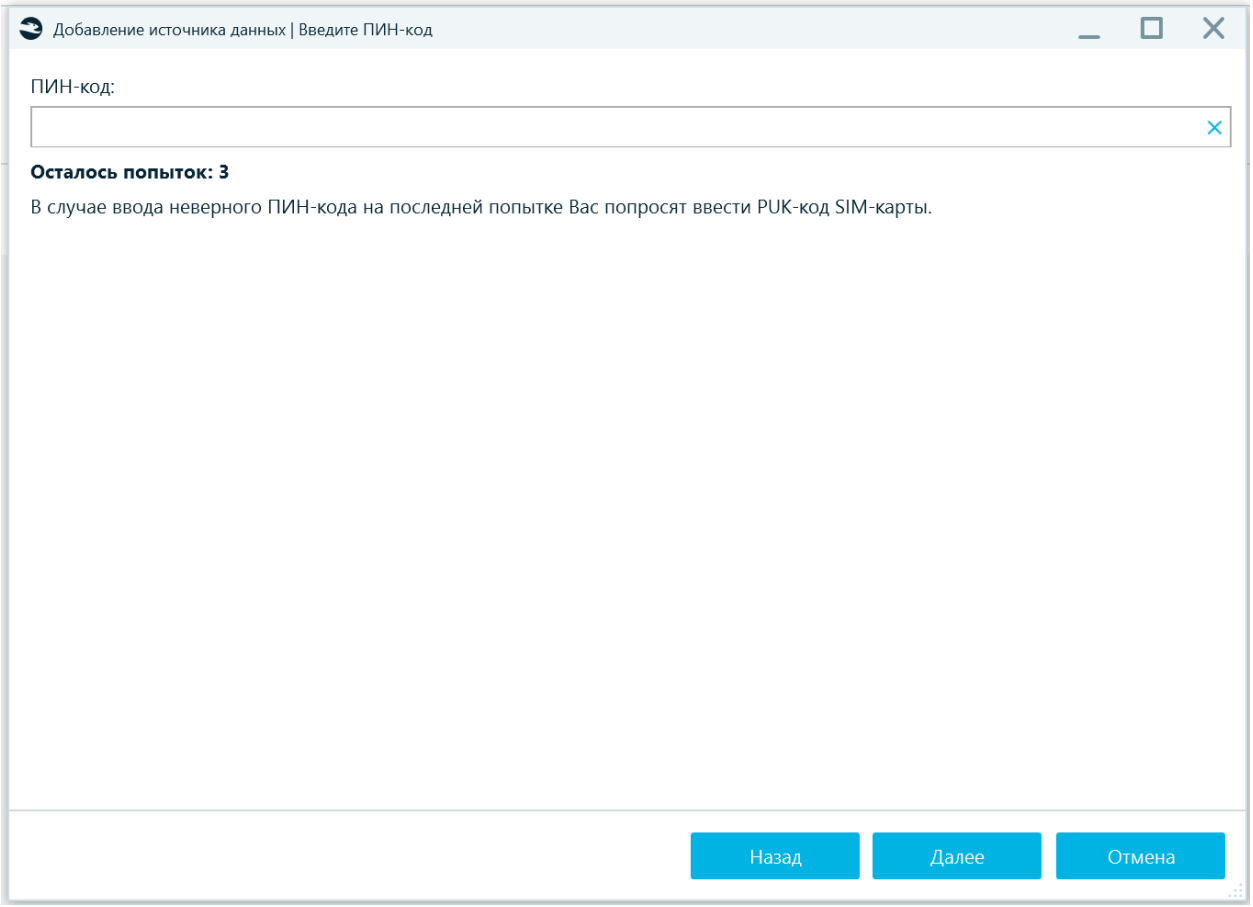

Рис. 8: Окно ввода PIN-кода.

При вводе неправильного кода пользователю будет выведено предупреждение о неверном коде (рис. 9), и количество попыток будет уменьшено на одну.

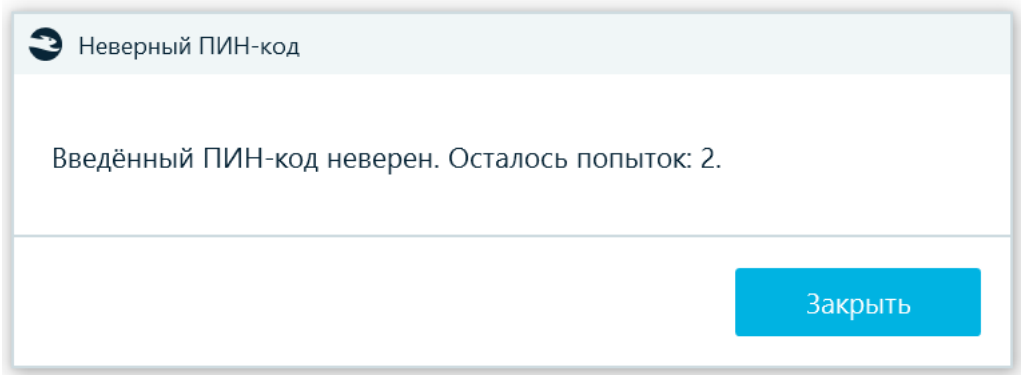

Рис. 9: Сообщение об ошибке подтверждения PIN-кода.

5. Особенности реализации

5.1. Реализация компоненты, взаимодействующей со считывателем карт

Разработанный модуль извлечения данных с SIM-карт, представленный на рис. 5 (диаграмма компонентов UML), состоит из нескольких частей. Первая часть, SimReaderAcquisitionHost, отвечает за выбор извлекаемых файлов и сохранение полученных данных в бинарные файлы. Вторая часть, SimFileSystemReader, занимается считыванием содержимого файловой системы SIM-карты. Третья часть, SimReader, отвечает за взаимодействие со считывателем карт. Взаимодействие между этими частями происходит через SimReaderHandler.

Для взаимодействия со считывателем карт необходимо установить обработчик на COM-порт, к которому подключён считыватель карт. Обработчик можно установить с помощью команды WinAPI CreateFileA[3]. После открытия обработчика необходимо прочитать ATR-байты — системные байты, содержащие информацию о состоянии SIM-карты [1].

Отправить команду на считыватель карт можно с помощью функции WinAPI WriteFile[4], а прочитать ответ считывателя карт с помощью функции WinAPI ReadFile[5]. Между отправкой команды и получением ответа необходимо подождать некоторое время.

Алгоритм извлечения данных SIM-карт представлен на рис. 10 (диаграмма последовательности UML).

[3]https://learn.microsoft.com/en-us/windows/win32/api/fileapi/nf-fileapi-createfilea
[4]https://learn.microsoft.com/en-us/windows/win32/api/fileapi/nf-fileapi-writefile
[5]https://learn.microsoft.com/en-us/windows/win32/api/fileapi/nf-fileapi-readfile

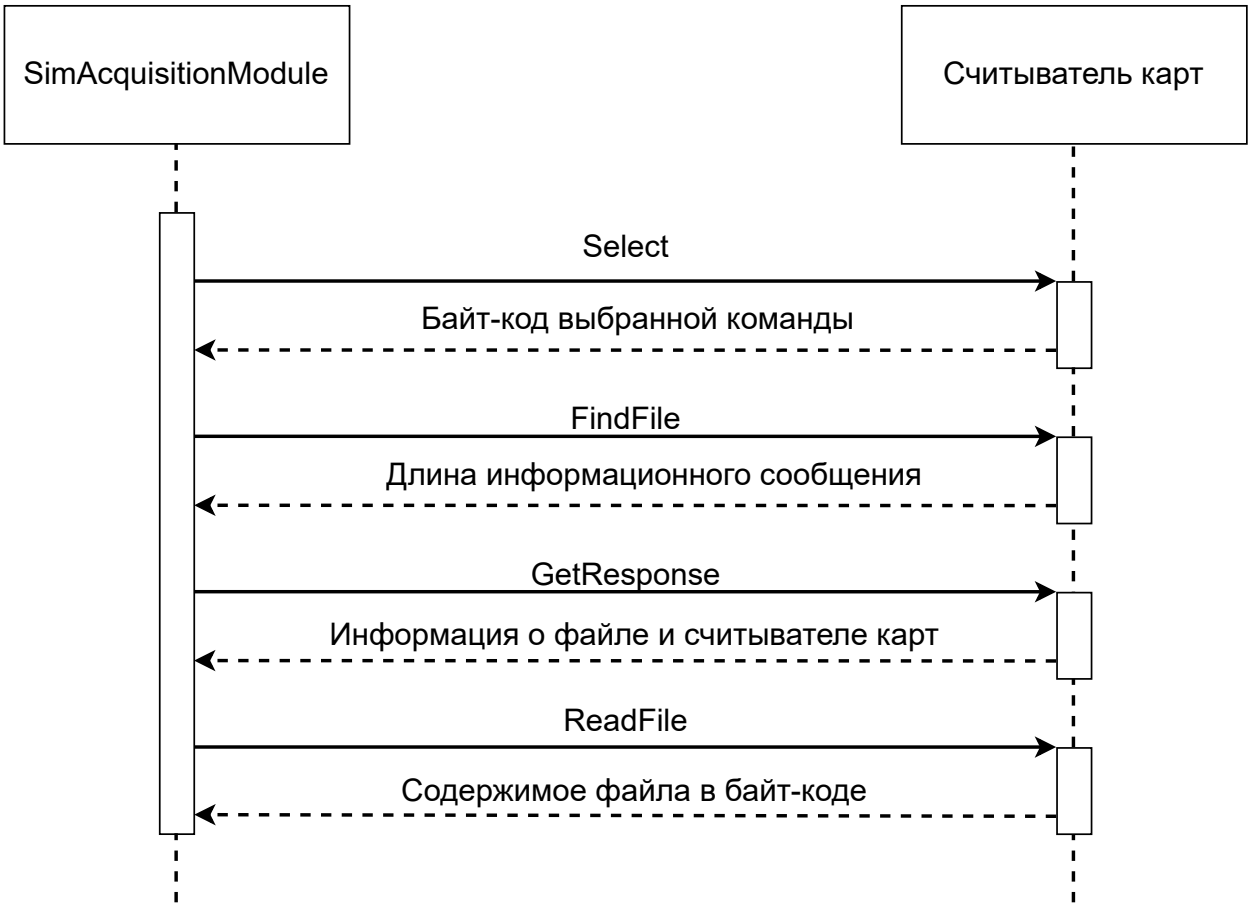

Рис. 10: Алгоритм чтения файла SIM-карты.

Для выбора файла для чтения на SIM-карте необходимо отправить несколько команд на считыватель карт. Первая команда Select предназначена для оповещения считывателя карт о необходимости выбрать файл. В этом запросе указывается специальный код команды, описанный в документации. В ответ считыватель карт повторяет отправленную команду в байтах, 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. Такие блоки также содержат длину считываемого файла.

Следующая команда — FindFile — направлена на поиск считываемого файла. В запросе 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 данных с информацией о считывателе карт и файле. Если файл не удалось найти, отправляются специальные байты.

Следующая команда — ReadInfoFile — направлена 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.

Последняя команда — ReadFile — направлена 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 для считывания файловой системы SIM-карты реализована на языке программирования C#. Её структура представлена на рис. 11 (диаграмма классов UML). Компонента SimFileSystemReader реализована с применением шаблона проектирования «Шаблонный метод».

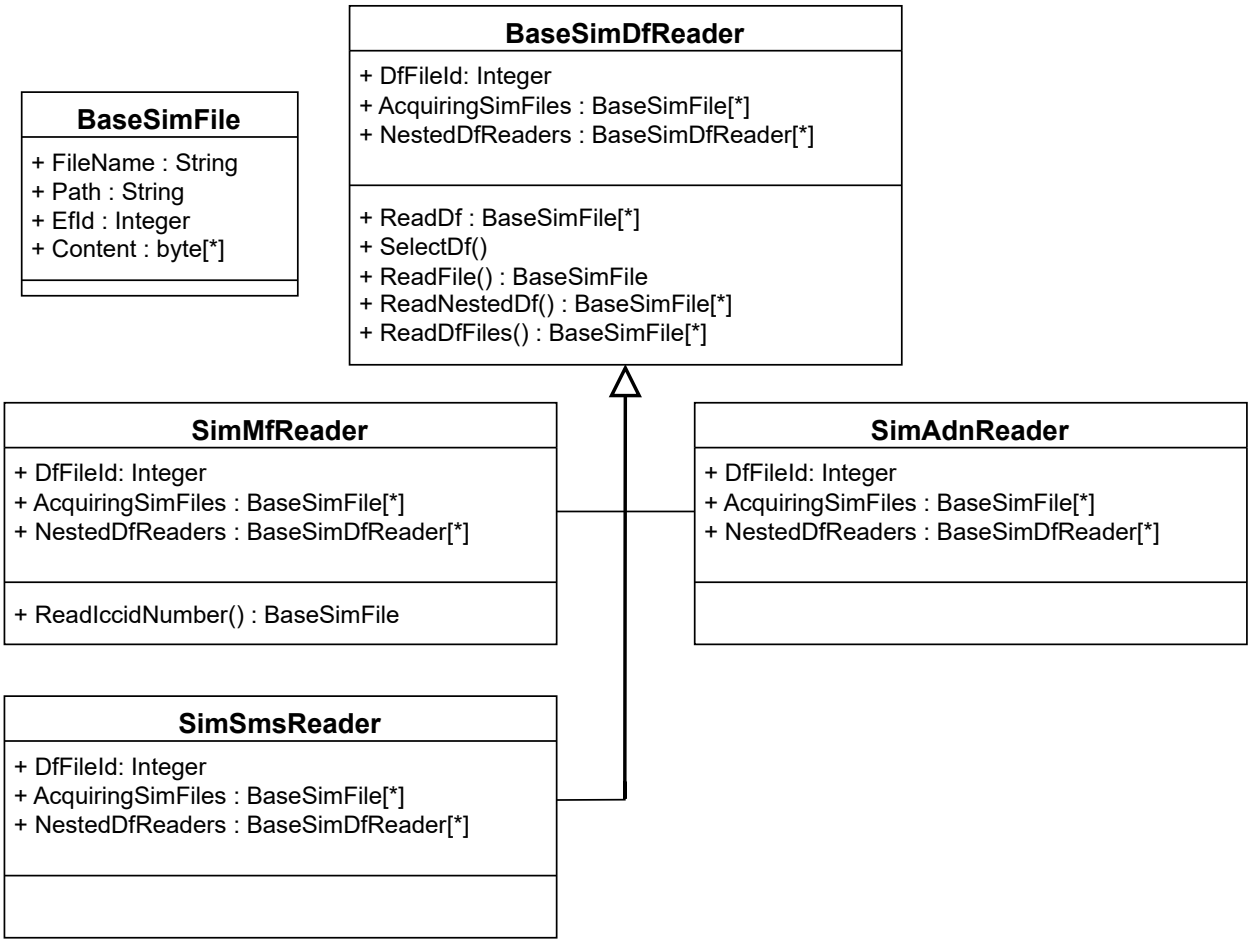

Рис. 11: Устройство компоненты, считывающей файловую систему SIM-карты.

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

Компонента для разбора извлечённой файловой системы SIM-карты также реализована на языке программирования C#. Реализация компоненты представлена на рис. 12 (диаграмма классов UML). Компонента SimDataParsingModule реализована с использованием шаблонов проектирования «Шаблонный метод» и «Фабричный метод».

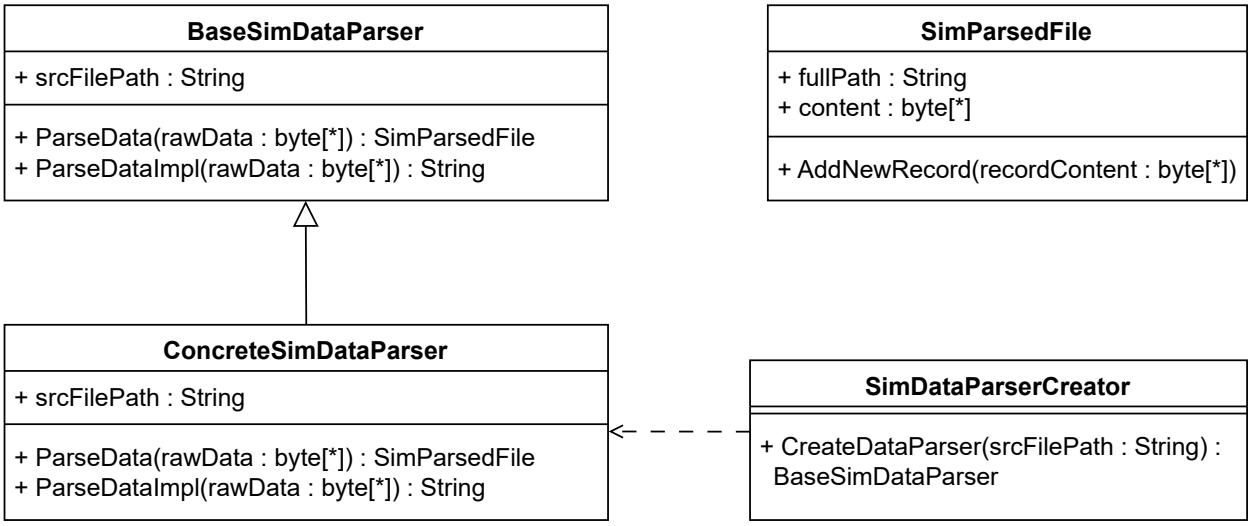

Рис. 12: Устройство компоненты, разбирающей файловую систему SIM-карты.

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

Фабричный метод реализован в классе SimDataParserCreator. Метод CreateDataParser(srcFilePath : String) создает экземпляр объекта 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 извлечения данных SIM-карты, а также продукт Belkasoft X, в который про-

изводится интеграция, реализованы на языках программирования C# и C++.

Чтобы использовать C++ код в C# проекте, была создана компонента SimReaderHandler. Она реализована на языке C++/CLI, что позволяет вызывать код, написанный на C++, в C#-проекте. Технология C++/CLI выбрана для упрощения отладки C++-кода из C#-проекта.

SimReaderHandler работает как «Фасад» и предоставляет только две функции: выбрать файл и считать его содержимое. Вся внутренняя логика скрыта в компоненте SimReader.

6. Тестирование и апробация

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

В сравнении с другим доступным продуктом реализованный модуль извлёк такое же число файлов SIM-карты. Тем не менее модуль, встроенный в Belkasoft X, выполнил разбор большего числа файлов, что может быть полезно для экспертов в области цифровой криминалистики.

Также в процессе тестирования было обнаружено, что операторы Tele2, Megafon и Beeline используют разные методы кодирования файлов на SIM-картах. Разработанный модуль смог успешно преобразовать текст для всех алгоритмов кодирования.

Реализованная функциональность прошла проверку кода, была интегрирована в исходный код проекта Belkasoft X и была проверена командой тестирования компании «Цифровая Корпоративная Защита». Метод извлечения данных SIM-карты с использованием считывателя карт доступен в официальной версии программы Belkasoft X от мая 2024 года.

7. Заключение

В ходе данной работы были получены следующие результаты.

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

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

Список литературы

[1] Answer To Reset (ATR), CardLogic. — URL: https://www.cardlogix.com/glossary/atr-answer-to-reset-smart-card/ (дата обращения: 12 мая 2024 г.).

[2] Belkasoft X Forensic, Belkasoft. — 2024. — URL: https://belkasoft.com/ru/x (дата обращения: 13 апреля 2024 г.).

[3] Charles Griffiths. The Latest 2024 Cyber Crime Statistics, AAG. — 2024. — 4. — URL: https://aag-it.com/the-latest-2022-cyber-crime-statistics/ (дата обращения: 13 апреля 2024 г.).

[4] Christopher Swenson Gavin Manes Sujeet Shenoi. Imaging and analysis of GSM SIM cards. — ResearchGate, 2005. — 2. — P. 4–5. — URL: https://www.researchgate.net/publication/45816123_Imaging_and_analysis_of_GSM_SIM_cards (дата обращения: 13 апреля 2024 г.).

[5] E3: Electronic Evidence Examine, Paraben Corporation. — 2024. — URL: https://paraben.com/ (дата обращения: 13 апреля 2024 г.).

[6] EF SST (SIM service table) // Digital cellular telecommunications system (Phase 2+); Specification of the Subscriber Identity Module Mobile Equipment (SIM-ME) interface (3GPP TS 51.011 version 4.14.0 Release 4) / ETSI. — 2005. — P. 39–41. — URL: https://www.etsi.org/deliver/etsi_ts/151000_151099/151011/04.14.00_60/ts_151011v041400p.pdf (дата обращения: 13 апреля 2024 г.).

[7] Gerard L. Pinto. osmo-sim-auth. — 2017. — 10. — URL: https://github.com/GerardPinto/osmo-sim-auth (дата обращения: 13 апреля 2024 г.).

[8] Gordon F. Snyder. What Information Can Be Pulled Off A Mobile Device SIM Card? — 2018. — 9. — URL: http://www.gordostuff.

com/2018/09/what-information-can-be-pulled-off.html (дата обращения: 13 апреля 2024 г.).

[9] How to Handle Data Acquisition in Digital Forensics, EC-Council Cybersecurity Exchange. — 2022. — 3. — URL: https://www.eccouncil.org/cybersecurity-exchange/computer-forensics/data-acquisition-digital-forensics/ (дата обращения: 13 апреля 2024 г.).

[10] Kamil Wartanowicz. SimLAB. — 2016. — 5. — URL: https://github.com/kamwar/simLAB (дата обращения: 13 апреля 2024 г.).

[11] Linux File System, Javatpoint. — URL: https://www.javatpoint.com/linux-file-system (дата обращения: 13 апреля 2024 г.).

[12] Michael Bosson. ICCID numbers and how to find them, onomondo. — 2023. — 6. — URL: https://onomondo.com/blog/iccid-number-explained/ (дата обращения: 13 апреля 2024 г.).

[13] Milan. What is SIM Card Reader and How Does it Work?, Hybrid Sim. — 2021. — 1. — URL: https://hybridsim.com/sim-card-reader/ (дата обращения: 13 апреля 2024 г.).

[14] Oxygen Forensics Detective, Oxygen Forensics. — 2024. — URL: https://www.oxygen-forensic.com/en/products/oxygen-forensic-detective (дата обращения: 13 апреля 2024 г.).

[15] Piotr Zerynger. DualSIMCard. — 2019. — 3. — URL: https://github.com/ITger/DualSIMCard (дата обращения: 13 апреля 2024 г.).

[16] Rozewski Manuel. What Information Is Stored On A SIM Card?, SimOptions. — 2021. — 8. — URL: https://www.simoptions.com/sim-card-information/ (дата обращения: 13 апреля 2024 г.).

[17] Serial Port Monitor. Track and analyze the activity of your COM ports, Electronic Team, Inc. — 2024. — URL: https://www.com-port-monitoring.com/ (дата обращения: 13 апреля 2024 г.).

[18] Smart Card Standards, QCard. — URL: https://www.q-card.com/about-us/smart-card-standards/page.aspx?id=1461 (дата обращения: 13 апреля 2024 г.).

[19] What Is Digital Forensics?, Simplilearn. — 2023. — 8. — URL: https://www.simplilearn.com/what-is-digital-forensics-article (дата обращения: 13 апреля 2024 г.).

[20] What is IMSI (International Mobile Subscriber Identity)?, SIMON IoT. — 2024. — 1. — URL: https://www.simoniot.com/what-is-an-imsi/ (дата обращения: 13 апреля 2024 г.).