УДК 004.42

Подставников Владислав Юрьевич
Амурский государственный университет
г. Благовещенск, Россия
E-mail: coololcat3@gmail.com
Фомин Денис Васильевич
Амурский государственный университет
г. Благовещенск, Россия
E-mail: gefest-uni@yandex.ru
Самохвалова Светлана Геннадьевна
Амурский государственный университет
г. Благовещенск, Россия
E-mail: sgs@amursu.ru
Podstavnikov Vladislav Yurievich
Amur State University
Blagoveschensk, Russia
E-mail: coololcat3@gmail.com
Fomin Denis Vasilyevich
Amur State University
Blagoveschensk, Russia
E-mail: gefest-uni@yandex.ru
Samokhvalova Svetlana Gennadievna
Amur State University
Blagoveschensk, Russia
E-mail: sgs@amursu.ru

ВИЗУАЛИЗАЦИЯ РАБОТЫ КРИПТОГРАФИЧЕСКИХ АЛГОРИТМОВ

VISUALIZATION OF CRYPTOGRAPHIC ALGORITHMS

Аннотация. Рассматриваются существующие криптографические алгоритмы и программы визуализации их работы. Обосновывается актуальность разработки программного инструмента визуализации работы криптоалгоритмов. Делается выбор инструментов разработки.

Abstract. The existing cryptographic algorithms and programs for visualizing their work are considered. The relevance of the development of a software tool for visualizing the work of crypto algorithms is substantiated. The choice of development tools is made.

Ключевые слова: криптография, комплексы программ, криптографические алгоритмы, автоматизация, кибербезопасность, информационная безопасность.

Key words: cryptography, software packages, cryptographic algorithms, automation, cybersecurity, information security.

DOI: 10.22250/20730268_2023_101_165

Введение

Современный мир невозможно представить без использования криптографии. Криптографические алгоритмы широко применяются в области кибербезопасности, защиты конфиденциальной информации и цифровых подписей. А также в других сферах – банковском деле, электронной коммерции, мобильных приложениях. Однако эффективное использование криптографических средств требует хотя бы минимального понимания основных принципов работы криптографических алгоритмов, их сильных и слабых сторон, пределов их эффективного применения [1, 2].

Современные криптоалгоритмы достаточно сложны для понимания. Разработка программного инструмента, визуализирующего их работу, позволит облегчить знакомство и освоение криптографии в ходе подготовки студентов как профильных, так и не профильных направлений среднего профессионального и высшего образования, включая обучающихся на курсах повышения квалификации и переподготовки. В свою очередь это позволит повысить общий уровень ИТ-специалистов, а также их грамотность в области кибербезопасности.

Криптографические алгоритмы широко используются для защиты данных от несанкционированного доступа и изменения. К примеру, шифрование позволяет защитить конфиденциальные данные (личные данные клиентов, кредитная информация и др.) [3].

Электронные цифровые подписи используются для подтверждения подлинности и целостности электронных документов и сообщений, что особенно важно при дистанционном обмене информацией. Электронные подписи позволяют убедиться в личности отправителя, а также в том, что полученное сообщение не было изменено в процессе передачи.

Аутентификация с применением криптографии позволяет удостоверить личность участников информационного обмена, что позволяет защитить от несанкционированного доступа персональные данные, а также дает возможность совершения юридически значимых действий онлайн (например, управление финансами и подача обращений в органы государственной власти).

Соответственно, криптография широко применяется в таких областях человеческой деятельности как банковское дело, электронная коммерция, мобильные приложения. В частности, криптографические алгоритмы используются для защиты финансовых транзакций и личной информации клиентов в банковской сфере. Криптография позволяет также защитить персональные данные клиентов, и чувствительную финансовую информацию, – например, данные кредитных карт, необходимые для совершения онлайн-платежей. Криптография применяется также для защиты данных в мобильных приложениях – таких как мессенджеры и социальные сети для защиты личной информации пользователей (сохранение тайны переписки) [2, 3].

Современные криптографические алгоритмы

В настоящее время криптография имеет в своем арсенале сложную систему алгоритмов, обеспечивающих защиту информации в контексте различных практических задач. Эта система алгоритмов включает в себя блочные, потоковые, асимметричные и симметричные алгоритмы. Каждый из них имеет свои преимущества и недостатки. Однако при разработке программного продукта невозможно сразу реализовать все существующие алгоритмы криптографической защиты информации. Поэтому стоит рассмотреть существующие группы, типы и конкретные алгоритмы и выбрать из них те, реализовать которые будет наиболее целесообразно.

Блочные алгоритмы представляют собой криптографические алгоритмы, работающие с блоками данных. Они используются для защиты информации, которая передается между двумя устройствами. В семействе блочных алгоритмов наиболее широко известны следующие алгоритмы.

TEA (Tiny Encryption Algorithm) – применяется для шифрования небольших блоков данных, обеспечивает хорошую скорость шифрования и дешифрования и является наиболее популярным алгоритмом данного семейства [2].

AES (Advanced Encryption Standard) – один из наиболее распространенных и широко используемых блочных алгоритмов, являющийся стандартом шифрования данных, составляющих государственную тайну (в США), а также использующийся для защиты информации в банковской, финансовой и многих других областях жизнедеятельности [2].

DES (Data Encryption Standard) – один из самых старых из известных блочных алгоритмов шифрования. Разработан в начале 1970-х гг., недолго но широко применялся для защиты данных во многих отраслях. Однако выявленные критические недостатки алгоритма свели его криптостойкость практически к нулю [2].

Blowfish – блочный алгоритм шифрования, разработанный в 1993 г. в качестве альтернативы DES. Однако даже сейчас этот алгоритм предоставляет высокую степень защиты информации [3].

Twofish – блочный алгоритм шифрования, разработанный в 1998 г. в качестве альтернативы AES, является продолжением и развитием идей криптоалгоритмов Blowfish, SAFER и SQUARE. До сих пор предоставляет достаточно высокую степень защиты информации [2].

Потоковые алгоритмы. Их отличительной способностью является нацеленность на шифрование потоков данных (на вход алгоритма подается не конкретный, ограниченный, конечный массив данных, а некоторая последовательность данных, причем неизвестно заранее, когда она закончится и какие данные поступят в следующий момент). Поэтому, как правило, криптоалгоритмы такого типа применяются для защиты передачи информации. В семействе потоковых алгоритмов широко известны следующие.

RC4 (Rivest Cipher 4) – один из самых известных потоковых шифров. Используется во многих протоколах безопасности компьютерных сетей – таких как SSL и WEP [2].

Salsa20 – потоковый шифр, обеспечивающий хорошую производительность на различных аппаратных платформах [4].

A5/1 и A5/2 – потоковые шифры, используемые в стандарте сотовой связи GSM для защиты передачи голоса и данных [2].

ChaCha20 – потоковый шифр, являющийся усовершенствованной версией Salsa20, обеспечивает высокую производительность и безопасность [3].

Асимметричные алгоритмы работают на основе использования открытого и закрытого ключей. При этом ключи одной пары не выводимы друг из друга и не взаимозаменяемы – у каждого из них своя роль в процессах шифрования и дешифровки. Они обеспечивают более высокий уровень безопасности, чем симметричные алгоритмы. Семейство асимметричных алгоритмов включает в себя следующий ряд алгоритмов.

RSA (Rivest-Shamir-Adleman) – один из самых распространенных алгоритмов, используемых в настоящее время, основан на сложности задачи факторизации больших целых чисел, используется как для непосредственного шифрования сообщений, так и в качестве элемента механизма цифровой подписи [4].

ElGamal – еще один алгоритм, основанный на математических задачах, связанных с теорией чисел, применяется для шифрования и цифровой подписи сообщений [4].

DSA (Digital Signature Algorithm) – алгоритм цифровой подписи сообщений, использующий математические примитивы из теории чисел и теории групп [2].

ECC (Elliptic Curve Cryptography) – семейство алгоритмов, основанных на сложности задачи вычисления дискретного логарифма на эллиптических кривых, используется для шифрования и цифровой подписи сообщений [2, 3].

Diffie-Hellman – протокол обмена ключами, используется для создания общего секретного ключа между двумя сторонами, не обменивающимися ключами заранее, основан на сложности задачи дискретной логарифмизации [1, 2].

Симметричные алгоритмы используют один и тот же ключ для шифрования и дешифрования данных. Семейство симметричных алгоритмов включает блочные и потоковые шифры. Говоря о наиболее простых и известных алгоритмах данного семейства, можно перечислить следующие группы алгоритмов [3].

Шифр замены – каждый символ в открытом тексте заменяется на другой символ в соответствии с определенным правилом замены. Простейший пример – шифр Цезаря, где каждая буква сдвигается на определенное число позиций в алфавите.

Шифр перестановки – символы в открытом тексте переставляются по определенному правилу, чтобы создать шифрованный текст. Простейший пример – шифр столбцов, где символы открытого текста записываются в строку, а читаются по столбцам.

Подводя итог рассмотрения существующих криптографических алгоритмов, можно сделать вывод о целесообразности включения в состав первой версии разрабатываемого программного продукта следующие алгоритмы: 1) ECDH; 2) TEA; 3) шифр Цезаря; 4) RC4.

Выбор этих алгоритмов обусловлен тем, что они представляют все основные семейства криптоалгоритмов. Что позволит пользователям программного продукта получить широкие базовые представления о криптографической защите информации. При этом стоит учесть, что выбранные алгоритмы имеют особенности, влияющие на их реализацию. Например, ECDH относится к алгоритмам асимметричного шифрования и используется для обмена ключами. Его преимущество в том, что он не подвержен атакам типа «человек посередине» и «атаки по времени». Кроме того, ECDH обеспечивает более высокую, чем другие асимметричные алгоритмы, скорость обмена ключами. Тем не менее, как и любой другой алгоритм, основанный на эллиптических кривых, ECDH достаточно труден для понимания.

TEA – блочный алгоритм шифрования, использующий ключевое расписание для шифрования и дешифрования блоков данных. Преимущество TEA в том, что он достаточно прост в реализации и может работать на многих аппаратных платформах. Однако он может быть подвержен атакам типа «выбранный текст» и «выбранный шифротекст» [4].

Шифр Цезаря – простой алгоритм шифрования, который использует сдвиг букв в алфавите, широко использовался в древности, но его безопасность для сегодняшнего дня очень низкая. Однако, он выбран для включения в состав программного продукта для демонстрации простейших базовых принципов работы криптографических алгоритмов [1].

RC4 – потоковый алгоритм шифрования, использующий ключевую последовательность для генерации псевдослучайных чисел с целью шифрования данных. Преимущество данного алгоритма

заключается в том, что он может работать на широком спектре различных аппаратных платформ и достаточно быстр в работе. Его недостаток – подверженность атакам типа «известный шифротекст» и «выбранный шифротекст» [2].

Существующие решения и способы визуализации работы алгоритмов

Разнообразие программных продуктов и инструментов, которые позволяют визуализировать работу криптографических алгоритмов, достаточно мало. Среди наиболее примечательных из них можно перечислить следующее программное обеспечение.

Crypto Playground – находится в стадии активной разработки. Планируется, что он будет позволять визуализировать работу различных криптографических алгоритмов, включая AES, DES, RSA, будет отображать каждый шаг алгоритма в виде текстовых сообщений, которые могут быть прочитаны и понятны только для тех, кто уже знаком с алгоритмом.

GCHQ CyberChef – позволяет создавать собственные «рецепты» для обработки данных, включая шифрование и дешифрование. При этом он предоставляет визуальный интерфейс для отображения и редактирования рецептов. Тем не менее, данный подход более интересен для энтузиастов-исследователей, чем для студентов, которым необходимо изучить конкретный алгоритм.

При анализе существующих решений и способов визуализации работы алгоритмов выявлено, что большинство из них не удобны и не просты в использовании. Некоторые не обеспечивают достаточной интерактивности и не дают возможности пользователям изменять параметры алгоритмов, другие могут оказаться слишком сложными для понимания, особенно начинающих пользователей.

Проведенный анализ различных способов визуализации работы криптографических алгоритмов дает возможность сделать выбор в пользу интерактивного отображения упрощенной блок-схемы алгоритма. Это позволит пользователям легче понимать, как работает алгоритм и какие шаги нужно предпринять для шифрования или дешифрования данных, а также легко и быстро изменять параметры алгоритма, наблюдая их влияние на его работу.

Использование интерактивной блок-схемы – удобный и понятный способ визуализации работы выбранных алгоритмов, применимый к блочным, потоковым и другим семействам алгоритмов. Кроме того, интерактивная блок-схема является достаточно простым и понятным инструментом.

Одновременно с этим для шифра Цезаря и некоторых «примитивных» алгоритмов возможна реализация визуализации на основе ряда «ассоциаций», индивидуальных для алгоритма. Для шифра Цезаря это лента/окружность с двумя алфавитами, сдвинутыми относительно друг друга.

Программный комплекс CryptoLearn

Исходя из вышеперечисленных фактов, имеет смысл разработать программный комплекс, позволяющий визуализировать криптографические алгоритмы. Цель такого комплекса – помочь в изучении основ криптографических алгоритмов. Предлагаемое название комплекса – «CryptoLearn».

Комплекс будет поддерживать работу с операционными системами Windows 7 и выше как на рабочих станциях, так и на серверной архитектуре, а также deb-based дистрибутивами Linux, в том числе AstraLinux.

Функционал программного комплекса будет включать в себя следующие возможности:

1) визуализацию работы выбранных криптографических алгоритмов в интерактивном режиме;

2) выдачу учащимся логинов и паролей для выполнения заданий на шифрование/дешифровку сообщений с помощью выбранных алгоритмов;

3) разнообразные форматы заданий, включающие в себя как шифровку/дешифровку сообщений, так и освоение алгоритмов;

4) оценку выполненных заданий и освоенности алгоритмов с помощью тестирования и занесение результатов в базу данных.

Для реализации программного комплекса будет использована клиент-серверная архитектура, позволяющая учителю контролировать выполнение заданий учащимися и проверять результаты.

Программный комплекс «CryptoLearn» будет создан на основе библиотек и компонентов, обеспечивающих высокую производительность и стабильность работы. В качестве языка программирования для реализации продукта предполагается использовать C#. Данный язык отличается удобством, поддержкой ООП-парадигмы, нацеленностью на разработку проектов такого масштаба, стабильностью работы и кроссплатформенностью.

Графический интерфейс пользователя предполагается разрабатывать с помощью библиотек .NET и Qt. Для функционала тестирования предполагается использовать СУБД SQLite.

Для C# существует несколько кроссплатформенных библиотек, позволяющих строить интерактивные блок-схемы. Например, NodeGraphControl – это библиотека, разработанная для платформы .NET, которая позволяет создавать графические пользовательские интерфейсы для работы с блок-схемами и узлами. NodeGraphControl имеет широкие возможности для создания пользовательских элементов и расширения интерфейса.

MindFusion.Diagramming for WinForms – библиотека для создания диаграмм и блок-схем на платформе .NET. Она имеет большое количество встроенных элементов и позволяет создавать пользовательские элементы, а также поддерживает редактирование и масштабирование.

GoJS – кроссплатформенная библиотека для построения интерактивных диаграмм, включая блок-схемы, на платформе .NET. Она предоставляет большое количество встроенных шаблонов и возможности для создания пользовательских элементов.

Для работы с шифрованием в C# также существует несколько библиотек. Например, BouncyCastle – это кроссплатформенная библиотека для работы с криптографическими алгоритмами на платформе .NET. Crypto++ – библиотека для работы с криптографическими алгоритмами на платформе .NET. OpenSSL – кроссплатформенная библиотека, поддерживающая шифрование и другие криптографические алгоритмы на платформе .NET.

Все перечисленные библиотеки отличаются быстродействием, стабильностью и наличием необходимого для данного программного продукта функционалом.

Заключение

Анализ различных типов криптографических алгоритмов, существующих программных продуктов визуализации их работы, а также инструментов разработки позволяет сделать вывод, что разработка программного комплекса «CryptoLearn» является актуальной задачей. Разрабатываемое программное обеспечение должно обладать главным образом следующими функциями: 1) визуализация блок-схем алгоритмов, 2) выполнение шифрования и дешифрования, 3) создание заданий для студентов.

Работа программного комплекса «CryptoLearn» может осуществляться на различных программно-аппаратных платформах, работающих под управлением операционных систем семейств Windows и Linux. Для реализации такого ПО можно использовать специализированные библиотеки готового программного кода, что позволит создать удобный и функциональный продукт.

Использование программного комплекса «CryptoLearn» должно привести к следующим положительным результатам:

1) улучшение качества обучения – благодаря возможности визуализации работы алгоритмов студенты смогут лучше понимать принципы работы шифрования и дешифрования данных, а также получать быструю обратную связь по выполненным заданиям;

2) экономия времени – использование программного комплекса позволит упростить процесс проверки заданий студентов, что сократит время, затрачиваемое преподавателями на проверку и оценку работ;

3) увеличение интереса к изучению темы – интерактивный подход к обучению и возможность самостоятельной работы над заданиями могут повысить мотивацию студентов к изучению криптографии;

4) увеличение качества знаний – благодаря возможности проверки выполненных заданий и получения обратной связи студенты смогут закрепить свои знания и улучшить свои навыки работы с криптографическими алгоритмами.

В дальнейшем развитие программного комплекса «CryptoLearn» может идти по двум направлениям: 1) расширение перечня доступных криптографических алгоритмов; 2) добавление технических возможностей (например, реализация веб-интерфейса, реализация поддержки плагинов и ряд других).

1. Бабаш, А.В. Криптографические методы защиты информации: учебник для вузов / А.В. Бабаш, Е.К. Баранова. – М.: КноРус, 2016. – 189 с.

2. Мао, В. Современная криптография: Теория и практика – М.: Вильямс, 2005. – 768 с.

3. Рябко, Б.Я. Основы современной криптографии для специалистов в информационных технологиях / Б.Я. Рябко, А.Н. Фионов – М.: Научный мир, 2004. – 173 с.

4. Фергюсон, Н. Практическая криптография /Н.Фергюсон, Б. Шнайдер. – М.: Диалектика, 2004. – 432 с.