Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Кафедра «Телекоммуникационные системы»
Специальность 6М071900 «Радиотехника, электроника и телекоммуникации»

ДОПУЩЕН К ЗАЩИТЕ
Зав. кафедрой
к.т.н., Шагиахметов Д.Р.
(ученая степень, звание, ФИО) (подпись)
«____»_____2014 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ
пояснительная записка

на тему: «Исследование методов повышения криптографической стойкости»

Магистрант Сыдыкова Н. С. _____ группа МТСп-12-2
(Ф.И.О.) (подпись)

Руководитель д.ф-м.н., профессор _____ Козин И.Д.
(ученая степень, звание) (подпись) (Ф.И.О.)

Технический консультант _____ _____
(подпись) (Ф.И.О.)

Рецензент _____ _____ _____
(ученая степень, звание) (подпись) (Ф.И.О.)

Консультант по ВТ к.х.н. ст. преподаватель _____ Данько Е.Т.
(ученая степень, звание) (подпись) (Ф.И.О.)

Нормоконтроль к.х.н. ст. преподаватель _____ Кудинова В.С.
(ученая степень, звание) (подпись) (Ф.И.О.)

Алматы, 2014

Некоммеческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Факультет «Радиотехники, электроники и связи»
Специальность 6М071900 «Радиотехники, электроники и телекоммуникации»
Кафедра «Телекоммуникационных систем

ЗАДАНИЕ
на выполнение магистерской диссертации

Магистранту Сыдыкову Нурболу Сериковичу
(фамилия, имя, отчество)

Тема диссертации: «Исследование методов повышения криптографической стойкости»
утверждена Ученым советом университета № 142 от «31» 10. 2013 г.
Срок сдачи законченной диссертации «30» мая 2014 г.
Цель диссертационной повышение криптографической стойкости алгоритма шифрования передаваемой информации

Перечень подлежащих разработке в магистерской диссертации вопросов или краткое содержание магистерской диссертации:
1) анализ существующих методов шифрования;
2) анализ существующих методыовповышения криптостойкости алгоритмов шифрования;
3) осуществить криптоанализ на примере алгоритма RSA (Rivest Shamir Adleman);
4) произвести оценку параметров повышения криптостойкости алгоритма RSA (Rivest Shamir Adleman).

Перечень графического материала (с точным указанием обязательных чертежей)

Рисунок 2.1 – Четыре раунда преобразований в алгоритме MD5
Рисунок 2.2 – Обобщенная схема итерационных криптоалгоритмов
Рисунок 3.2 – Сгенерированный MD5-шифр сообщения
Рисунок 3.4 – Текстовый файл с хэш-шифрами
Рисунок 3.6 – Выбор параметров импортируемых данных
Рисунок 3.8 – Основное окно программной среды CrypTool 2 Beta
Рисунок 3.10– Функциональная схема процесса шифрования RSA
Рисунок 3.11 – Расчет ключа с параметрами 1
Рисунок 3.16 – Шифрование с применением хеш-функции MD5

Рекомендуемая основная литература:

1. Баричев С. Г., Гончаров В. В., Серов Р. Е. Основы современной криптографии. — М.: Горячая линия — Телеком, 2002.

2. http://www.rsa.com/rsalabs/node.asp?id=2253 – Статья «What are MD2, MD4, and MD5?» - RSA Laboratories, 2000.

3. Щербаков А., Домашев А. Прикладная криптография. М.: Русская редакция, 2003. 416 с.

4. Schieber R. Site Security Using Microsoft's CryptoAPI. N.Y.: Wrox, 2001. 49 p.

5. IP Authentication using keyed MD5 by Bart Preneel (ESAT, K.U.Leuven, Belgium) Bart.Preneel@esat.kuleuven.ac.be

6. MD5 vs SHA-1, Performance and Pedigree by Masatake Ohta mohta@necom830.hpcl.titech.ac.jp

7. MD5 vs SHA by Ryan Malayter rmalayter@bai.org

Г Р А Ф И К
подготовки магистерской диссертации

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
1 Информационный обзор согласно темы диссертации	01.10.2012	15 %
2 Создание модели для проведения эксперимента на базе программы Cryptool 2	14.03.2013	15 %
3 Экспериментальное исследование повышение криптостойкости алгоритма RSA	02.10.2013	20 %
4 Исследование влияния разных типов хеш-функций на стойкость алгоритма RSA	20.02.2014	15 %
5 Анализ полученных экспериментальных и расчетных данных	02.04.2014	20 %
7 Оформление диссертационной работы	25.05.2014	15 %

Дата выдачи задания 05.09.2012 г.

Заведующий кафедрой _____ Шагиахметов Д.Р.
(подпись) (Ф.И.О.)

Руководитель диссертации _____ Козин И.Д.
(подпись) (Ф.И.О.)

Задание принял к исполнению
магистрант _____ Сыдыков Н.С.
(подпись) (Ф.И.О.)

Аңдатпа

Осы магистрлік диссертация шифрлік алгоритмдерінің криптографиялық тұрақтылығын көтеру әдістерін зерттеуге арналады.

Зерттудің өзектілігі - құпия шифрдің жүйесінің тез дамуын, сондай-ақ, оларды бұзу әрекетінің қатар дамуы аса жоғары криптобұрақтылығы бар жаңа жүйені құруды талап етеді.

Зерттеу жұмысына RSA алгоритмы алынған және Cryptool 2 бағдарламалық өнімнің комегімен жасалды, сондай-ақ, әртүрлі әдістер бойынша шабуыл жүзеге аырылды.

Зерттеу жұмыстарының нәтежесі бойынша келесі тұжырым жасалды – хеш-қызметін қолдану арқылы алгоритмдерді сенімді, сондай-ақ, құпия шифрлар кілтінің өлшемдерін көбейту талап етімді.

Аннотация

Данная магистерская диссертация посвящена исследованию методов повышения криптографической стойкости алгоритмов шифрования.

Актуальностью исследования заключается в том, что быстрое развитие систем шифрования, а так же сопутствующее развитие их взлома ведет к созданию новых систем с более высокой криптостойкостью.

Для исследования был выбран алгоритм RSA и произведено моделирование его алгоритма с помощью программного продукта Cryptool 2, а так же осуществлена атака по различным методам.

По результатам исследования можно сделать вывод, что надежнее использовать алгоритмы с применением хеш-функций, а так же увеличить параметры ключей шифрования.

Annotation

This master's dissertation is devoted to research methods for improving the cryptographic strength of encryption algorithms.

Relevance of the study is that the rapid development of encryption systems, as well as the concomitant development of their breaking leads to the creation of new systems with higher cryptographic strength.

The RSA algorithm was selected for the study. This algorithm was modeled using software Cryptool 2, also it was carried out attacks on various methods.

According to the study it can be concluded that it is more reliable to use algorithms with hash functions, as well as increasing the options of the encryption keys.

Содержание

Введение

На сегодняшний день среди методов повышения криптостойкости системы шифрования можно отметить алгоритмы, основанные на хеш-функциях, а так же симметричные и ассиметричные алгоритмы. Вопросам исследования их криптостойкости отводится большое внимание.

В данной магистерской диссертации исследованы вопросы криптостойкости на примере алгоритма шифрования RSA с помощью программной среды CrypTool 2.

Программа использует вектор ориентированный графический интерфейс, основанный на Windows Presentation Foundation (WPF), что для пользователя уже является знакомым и интуитивно понятным, а также дает возможность масштабирования рабочего окна. На рисунке 1 приведено изображение основного окна программы.

Cryptool 2 Beta поддерживает множество известных от классических до современных видов шифрования (ассиметричные и симметричные) таких как Виженер, Цезарь, DES, AES, RSA. CrypTool 2 Beta предоставляет различные крипто аналитические средства для анализа или даже взлома классических и современных шифров (наиболее простейший способ перебора ключей и тд).

Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д.

Все шифры и дополнительные устройства, используемые в программе, сделаны в виде функционал блоков. Имеющие модуль настройки определенных соответствующих ей параметров для этого CrypTool2 предоставляет графический пользовательский интерфейс для визуального программирования. Тот же компонент может визуализировать свои внутренние операции. Это делает его удобным для пользователя, чтобы он мог проследить все детали криптографического алгоритма и увидеть более широкую картину того, какой сценарий использует этот шифр или блок в реальной жизни.

Для исследования был выбран алгоритм RSA и произведено моделирование его алгоритма с помощью программного продукта Cryptool 2.а так же осуществлена атака по различным методам.

1 Классификация методов шифрования

1.1 Основные понятия криптологии

Криптология разделяется на два основных направления - криптографию и криптоанализ. Цели этих двух направлений прямо таки противоположны. Криптография занимается поиском и исследованием математических методов преобразования информации.

Сфера таких данных интересов это исследование такой возможности расшифровывания информации без знания ключей.

Современная криптография включает в себя четыре крупных раздела:

- симметричные криптосистемы;
- криптосистемы с открытым ключом;
- системы электронной подписи;
- управление ключами.

Основные же направления использования таких данных криптографических методов - это быстрая передача само, что ни на есть, конфиденциальной информации по таким каналам связи (например, электронная почта), которые установление подлинности передаваемых сообщений, а так же хранение информации (документов, баз данных) на разных носителях в зашифрованном виде.

При такой быстрой обработке данных или хранится насамом компьютере, либо в памяти и различных типов информации [6] как способ предотвращения несанкционированного линий доступа связи. Криптографическое изменения блокировать передачу информации между различными элементами системы, автоматизированных систем, таких как защита данных и информационной безопасности криптографические методы могут быть использованы с долгой историей. В настоящее время они используются в теории и практике формы основе другого метода шифрования, есть несколько проблем. Большинство из этих методов может быть успешно использован для закрытия информации. Асимметричные методы шифрования Довольно распространенной формой RSA. Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д.

Так что читайте криптографическую (снижение) является единственным знание является ключом, который может позволить для преобразования данных.

Информация как конфиденциальная и вернется [6] Некоторые тексты строятся на алфавите. Согласно этим положениям, состоит в следующем.

Алфавит - конечное множество используемых для кодирования информации знаков.

Текст - упорядоченный набор из элементов алфавита.

В качестве примеров алфавитов, используемых в современных ИС можно привести следующие:

– алфавит Z_{33} - 32 буквы русского алфавита и пробел;

– алфавит Z256 - символы, входящие в стандартные коды ASCII и КОИ-8;

– бинарный алфавит - Z2 = {0,1};

– восьмеричный алфавит или шестнадцатеричный алфавит.

Шифрование - преобразовательный процесс: исходный текст, который носит также название открытого текста, заменяется шифрованным текстом.

Дешифрование - обратный шифрованию процесс. На основе ключа шифрованный текст преобразуется в исходный [6].

Ключ - информация, необходимая для беспрепятственного шифрования и дешифрования текстов. Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д.

Криптографическая система представляет собой семейство **T** преобразований открытого текста. Члены этого семейства индексируются, или обозначаются символом k; параметр k является ключом. Пространство ключей K - это набор возможных значений ключа. Обычно ключ представляет собой последовательный ряд букв алфавита.

Разделенные на симметричных криптосистем и открытого ключа. Так что читайте криптографическую (снижение) является единственным знание является ключом, который может позволить для преобразования данных.

В симметричных криптосистем и шифрования и дешифрования с использованием того же ключа.

В открытых ключей системы используют два ключа - государственные и частные, математически связаны друг с другом. Данные шифруются с помощью открытого ключа, который доступен всем, и расшифровывается с помощью закрытого ключа, известного только получателю сообщения. Довольно распространенной формой асимметричного метода шифрования является RSA.

Термины распределение ключей и управление ключами относятся к процессам системы обработки информации, содержанием которых является составление и распределение ключей между пользователями.

Электронной (цифровой) подписью называется присоединяемое к тексту его криптографическое преобразование, которое позволяет при получении текста другим пользователем проверить авторство и подлинность сообщения.

Криптостойкостью называется характеристика шифра, определяющая его стойкость к дешифрованию без знания ключа(т.е. криптоанализу). Имеется несколько показателей криптостойкости, среди которых:

– количество всех возможных ключей;

– среднее время, необходимое для криптоанализа.

Преобразование T_k определяется соответствующим алгоритмом и значением параметра k. Эффективность шифрования с целью защиты информации зависит от сохранения тайны ключа и криптостойкости шифра. Так что читайте криптографическую (снижение) является единственным знание является ключом, который может позволить для преобразования данных.

Процесс криптографического закрытия данных может осуществляться как программно, так и аппаратно. Аппаратная реализация отличается существенно большей стоимостью, однако ей присущи и преимущества: высокая производительность, простота, защищенность и т.д. Программная реализация более практична, допускает известную гибкость в использовании. Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д.

1.2 Требования к криптосистемам

Для современных криптографических систем защиты информации сформулированы следующие общепринятые требования:

– зашифрованное сообщение должно поддаваться чтению только при наличии ключа;

– число операций, необходимых для определения использованного ключа шифрования по фрагменту шифрованного сообщения и соответствующего ему открытого текста, должно быть не меньше общего числа возможных ключей;

– число операций, необходимых для расшифровывания информации путем перебора всевозможных ключей должно иметь строгую нижнюю оценку и выходить за пределы возможностей современных компьютеров (с учетом возможности использования сетевых вычислений);

– знание алгоритма шифрования не должно влиять на надежность защиты;

– незначительное изменение ключа должно приводить к существенному изменению вида зашифрованного сообщения даже при использовании одного и того же ключа;

– структурные элементы алгоритма шифрования должны быть неизменными;

– дополнительные биты, вводимые в сообщение в процессе шифрования, должен быть полностью и надежно скрыты в шифрованном тексте;

– длина шифрованного текста должна быть равной длине исходного текста;

– не должно быть проостых и легко устанавливаемых зависимостью между ключами, последовательно используемыми в процессе шифрования;

– любой ключ из множества возможных должен обеспечивать надежную защиту информации;

– алгоритм должен допускать как программную, так и аппаратную реализацию, при этом изменение длины ключа не должно вести к качественному ухудшению алгоритма шифрования [6].

Криптоанализ может производиться в совокупности с шифрованием. Так что читайте криптографическую (снижение) является единственным знание является ключом, который может позволить для преобразования данных. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д.

1.3 Симметрические криптосистемы

Традиционный долгосрочная схема, ключевым схема криптографической симметричны. В этой схеме ключ участвует в шифрования и расшифровки данных. Ключ процедура шифрования с использованием источника данных активностью в отношении ряда тем же ключевого кода, чтобы вернуть порядок, чтобы использовать влияние производится. Расшифровка кода есть без ключа возможно. Зашифрованные данные передаются как обычно, то есть, незащищенные каналы связи, и тот же ключ, что повышает уязвимость системы, создать потребность в дополнительном обмена ключами защищенному каналу, и отправитель и получатель должен быть готов, и должны увеличить организационные проблемы. Так что читайте криптографическую (снижение) является единственным знание является ключом, который может позволить для преобразования данных.

RSA асимметричные алгоритмы шифрования, довольно распространенным типом является путь.

Его размер велик и вся кодирование правило, вычислительный блок не развита, чистая процентов любой длины и разделен на блоки фиксированного длины, каждый входной блок, независимо от его положения в порядке, шифруется отдельно. Такое криптосистем систем шифрования, называется блоков. На практике, как правило, два широко используется шифрование смешивания дисперсии и принципы. Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д. Дисперсия чистом зашифрованных символов распространением влияния: Если вы нажмете сетевые статистические свойства. Этот принцип, развитие ключевых подразделений восстановления. Так что читайте криптографическую (снижение) является единственным знание является ключом, который может позволить для преобразования данных.Устраняет являются криптограммы отличным подарком для распространения влияния одной из основных характеристик этого знака. Связь и восстановление общественного зашифрованный устранить статистические характеристики использования шифрования, такого смешивания преобразования. Примирение хорошее общее

метод достижения распространение и использование последовательности осуществляется в соответствии с кодом, они составляют менее общей дисперсии и значительное пропаганды. Это наиболее распространенные алгоритмы шифрования для простых замен и перестановок. Изменения в существующих криптографических методов может также уменьшить следующим образом:

Многоалфавитная подстановка - кроме самой простой форме более или менее сложных изменений в правилах (по аналогии с белым) является изменение текстовых символов источника. Требовать использования высокого большим, чтобы обеспечить криптографические ключи.

Перестановки - простые способы криптографической. Часто используется в комбинации с другими методами. Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д. Так что читайте криптографическую (снижение) является единственным знание является ключом, который может позволить для преобразования данных.

Гаммирование - это метод, ключ псевдослучайной последовательности генерируется исходный код на основе используемого метода. Асимметричная алгоритм шифрования является довольно распространенным типом образом.

Блочные шифры зашифрованный текст блока (ов), используемых в основных методов преобразования (могут возникнуть в переходный период) представляет собой последовательность. Блочные шифры на практике, в связи с высокой надежностью класса "чистая" является более распространенным интерпретация, но и метод вы разбит на блоки, и практически любой длины позволит вам зашифровать текст. Шифры на основе классификации, Россия и стандарт шифрования США [6].

1.3.1 Метод Цезаря

Метод Цезаря является самым простым вариантом шифрования.

Он назван по имени римского императора Гая Юлия Цезаря, который поручал Марку Туллию Цицерону составлять послания с использованием 50-буквенного алфавита, сдвигая его на 3 символа вперед. Так что читайте криптографическую (снижение) является единственным знание является ключом, который может позволить для преобразования данных. Достаточно распространенным алгоритмом шифрования ассиметричного типа является метод. Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д.

Подстановка определяется по таблице замещения, содержащей пары соответствующих букв “исходный текст – шифрованный текст”.

Например, ВЫШЛИТЕ_НОВЫЕ_УКАЗАНИЯ посредством подстановки преобразуется в еюыолхиврсеюивцнгкгрлб.

Таблица 1.1 – Шифрование по методу Цезаря

А→г	Й→м	Т→х	Ы→ю
Б→д	К→н	У→ц	Ь→я
В→е	Л→о	Ф→ч	Э→_
Г→ж	М→п	Х→ш	Ю→а
Д→з	Н→р	Ц→щ	Я→б
Е→и	О→с	Ч→ъ	_→в
Ж→й	П→т	Ш→ы	
З→к	Р→у	Щ→ь	
И→л	С→ф	Ъ→э	

Система проста и чувствительна. Если злоумышленник

1) и соответствующий зашифрованный исходный код;

2) текст выбранной зашифрованного текста,

ключевые и исходные расшифровки определениями Кода являются стандартными.

Эта система использовала грубую силу, используя современные компьютерные технологии, это не слишком трудно взломать. Таким образом, способ криптографической нет. Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д.

И Хилл шифр шифр Pleyfera - более выгодным обобщения подстановки Цезаря. Они основаны на более чем одной замены символов и 2-г (код Pleyfera), или N-грамм (Хилл шифр). Высоко существенным, когда они реализуют криптографический ключ требуется относительно большое количество сложной информации.

1.3.2 Системы шифрования Вижинера

Метод Вижинера является следствием подстановки Цезаря. В системе Вижинера задается некая конечная последовательность ключа:

$$k = (k_0 ,k_1 ,...,k_n), \quad (1.1)$$

которая называется ключом пользователя, она продолжается до бесконечной последовательности, повторяя цепочку. Таким образом, получается рабочий ключ.

Например, при ключе пользователя 15 8 2 10 11 4 18 рабочий ключ будет периодической последовательностью:

15 8 2 10 11 4 18 15 8 2 10 11 4 18 15 8 2 10 11 4 18 ...

Таким образом:
При длине пользовательского ключа R
1) исходный текст x делится на R фрагментов:

$$x_i = (x_i , x_{i+r} , ..., x_{i+r(n-1)}), 0 \leq i < r; \quad (1.2)$$

2) i-й фрагмент исходного текста x_i шифруется при помощи подстановки Цезаря в зависимости от пользовательского ключа:

$$(x_i , x_{i+r} , ..., x_{i+r(n-1)}) \cdot (y_i , y_{i+r} , ..., y_{i+r(n-1)}). \quad (1.3)$$

Ключевое слово или фраза, таких как К за политику конфиденциальности, плохие, очень прост в использовании = (К0, К1, ..., КК-1), это было легко запомнить. Информационная система способна распознавать информационной безопасности. Так что читайте криптографическую (снижение) является единственным знание является ключом, который может позволить для преобразования данных. Ключи, аппаратное и программное обеспечение, чтобы получить использовать случайную генерацию ключей. Асимметричная алгоритм шифрования является довольно распространенным типом образом. Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д.

Преобразуем текст с помощью подстановки Вижинера (r=4):
Исходный текст (ИТ1):

НЕ_СЛЕДУЕТ_ВЫБИРАТЬ_НЕСЛУЧАЙНЫЙ_КЛЮЧ

Ключ: КЛЮЧ

Разобьем исходный текст на блоки по 4 символа:

НЕ_С ЛЕДУ ЕТ_В ЫБИР АТЬ_ НЕСЛ УЧАЙ НЫЙ_ КЛЮЧ

и наложим на них ключ (используя таблицу Вижинера):

Н+К=Ч, Е+Л=Р и т.д.

Получаем зашифрованный (ЗТ1) текст:

ЧРЭЗ ХРБЙ ПЭЭЩ ДМЕЖ КЭЩЦ ЧРОБ ЭБЮ_ ЧЕЖЦ ФЦЫН

Криптостойкость метода резко убывает с уменьшением длины ключа.

Тем не менее такая система как шифр Вижинера допускает несложную аппаратную или программную реализацию и при достаточно большой длине ключа может быть использован в современных ИС. Так что читайте криптографическую (снижение) является единственным знание является ключом, который может позволить для преобразования данных.

1.3.3 Гаммирование

Гаммирование является общим изменения криптографические процесса. Ключи гаммирование и использование шифрования огромные значения, а также условный симметричные шифры во время границы между типа Vizhinera.

Шифрования гаммирование принцип системного процесса датчика, случайное число шифр все открытые на спине, является создать приходит после гамма введения данных [6]. Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д.

Такой ключ конкретной информации расшифровки известны и некоторые информация шифруется таким воссоздание введения небольшого количества, чтобы уменьшить процесс шифрования. Асимметричное шифрование довольно распространенной формой. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных.

Количество содержать гамма-битный повторяющаяся последовательность, в результате, очень трудно расширить зашифрованы. Время любая часть исходного кода, который не знает сферу общей протяженностью зашифрованного текста является больше вычислительной мощности вы (Разбивка под ключ) можно записать только в этой области, каждый из которых имеет различное случайное слово зашифрованный код будет. В этом случае единственные пределы ключ и алгоритм криптографической. Асимметричное шифрование Довольно распространенной формой.

Злоумышленник знаменитый крах текста и криптограммы, по крайней мере метода гаммирования бесполезно. Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д. Субсидии и модульная процесс и приводит к снижению емкости всей восстановления последовательность шифра. Злоумышленник может сделать, основываясь на предположениях о содержании текста отображается. Большинство сообщений, отправленных словами "очень близко", а затем весь текст начинается гораздо проще криптоанализа является неравные алгоритмы шифрования довольно

распространенный тип. Очень реально, информационная безопасность всегда следует учитывать при создании системы [6].

1.4 Криптосистемы с открытым ключом

В 1976 г. У.Диффи и М.Хеллманом предложили совершенно новый тип криптографической системы – это система с открытым ключом [public keycryptosystem]. В такой схеме с наличием открытого ключа имеется сразу два ключа, открытый [public] и секретный [private, secret], которые выбраны так, что их последовательное использование, применив к массиву шифруемых данных, оставляет данный массив без каких-либо изменений. Процедура шифрования использует только открытый ключ, для дешифрования используется секретный ключ. Достаточно распространенным алгоритмом шифрования ассиметричного типа является метод. Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д. Неизвестный секретный ключ значение не представляется возможным, код Процесс расшифровки; Открытый ключ известен значение секретного ключа вычислительной невозможной работу. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных с помощью открытого ключа криптографического ключа преимущество - это простой механизм для обмена ключи, созданные. Может проводиться только в качестве открытых отношений каналов связи позволяет использовать обычные открытые каналы для дальнейшего открытого ключа, а также предоставляет всю информацию, необходимую для обеспечения специальных каналов передачи через значение ключа общественности.

Улучшенное использование открытого ключа защиты информации, и с такой системой, и, с криптографических функций. Она обеспечивает надежное шифрование информации, передаваемой к главной цели сегодня был использовать криптографический поле перед процессом лицензирования, и нотариально заверен (сертифицирован), распределенное управление, и голосование систему, так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных, электронные криптографических систем денежных и цифровые подписи (проверки подлинности) и все больше и введите. использование криптографии с открытым ключом системы, наиболее общей характеристикой шифрования, цифровых подписей, цифровых подписей, использовать традиционные шифрование недавно, по сравнению с чистым использования ключа системы общественного цифровой подписи, некоторые увеличились, но не поддерживает процесс шифрования. Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по

определенным стандартам или использование булевой алгебры и т.д. Асимметричное шифрование Довольно распространенной формой.

Цифровая подпись используется для идентификации пустыми сети связи, передающий тексты. Это похоже на нормальное знак рукой, и его основными характеристиками являются: человек, который был помещен в подписанном тексте подтверждает, что он идет, и долг народа, связанных с зашифрованным текстом подписан не позволяют уйти. Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д. Цифровые подписи передается кодированный текст, так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных, подписанный небольшого количества дополнительной информации. Только открытый ключ, закрытый ключ, и процесс шифрования для создания цифровой подписи с использованием другого теста.

Для высокоскоростного блока обработки данных для них такой же длины, симметричный ключ на основе открытой системы обычно в десять раз ниже, чем с алгоритмом цифровой подписи, шифрования и открытый ключ эти характеристики. С открытым ключом методы шифрования для повышения эффективности с использованием комбинации в ширину и использовать два вида криптографических алгоритмов. Симметричные ключевые данные с выбранного случайным симметричным ключом алгоритма, известного как исходный код, а затем шифрования симметричного ключа с помощью открытого ключа алгоритма шифрования, неравенство шифрования Довольно распространенной формой. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных. Открытый канал зашифрован симметричный ключ и симметричный ключ Открытый ключ передается в зашифрованном виде. В результате зашифрованного канала, а затем частной симметричным ключом и симметричного ключа, чтобы вернуться, чтобы вернуться и повернуть вспять процесс с помощью ключа. Электронные подписи от всего текста, который является весь текст короткий срок, и его хэш-функция способ (поглощается) [один хэш-функция вычисляет поглощенную знак; Его выбор хэш или почти неразрешимой хэш-функции является тот же текст в другом восстановления текста. Асимметричная алгоритм шифрования является довольно распространенным типом образом. Тем не менее, секретный ключ хэш-функции можно использовать только на текстовый блок, чтобы быть зашифрованы секретные ключевые цифровых подписей и шифрования в процессе создания; Канал создан текст и кэш цифровой подписи. Этот тест является цифровая подпись, цифровая подпись для хэш-функции зависит от значения открытого ключа для проверки, что процесс передачи передачи зашифрованного текста с использованием любого открытого канала рассматривается как хэш-функции. Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и

другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д. Методы для измерения таких как однонаправленных хэш-функций, как правило, очень тесно связана с симметричных алгоритмов шифрования с использованием ключей.

Описанные комбинированные методы шифрования и электронно-цифровой подписи объединяют в себе эффективность данных алгоритмов с использование симметричного ключа и свойство независимости от дополнительных секретных каналов передачи, присущее таким алгоритмам, которые используют открытые ключи. Криптографическая же стойкость конкретного комбинированного метода шифрования информации определяется стойкостью слабейшего звена в цепи, состоящей из алгоритмов с использованием симметричного и открытого ключей, выбранных для его реализации.

1.4.1 Система RSA

RSA асимметричные алгоритмы шифрования, довольно распространенным типом является путь. Этот подход в 1977-78, Рональд Ривест (R.Rivest), Ади Шамир (A.Shamir) и Леонардо Adlmanom (L.Adleman) предлагается. Эффективно используя секретный ключ с односторонних функций, они смогли выдвинуть свои собственные. RSA алгоритм Стойкость очень просто целые числа, так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных, факторинг основывается на сложности процесса. Алгоритмы факторизации, те же данные (факторинг), современное государство дает нам возможность разрешить этот вопрос в течение ряда 430-бит; Очевидно надежность - на основе, 512-битный длина ключа, чтобы пройти через 10 лет данных, и 1024 бит достаточно надежны, чтобы защитить. Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д. Нет математическая прозрачным проблема факторинга RSA не свести задачу, система будет одобрена, но критерии для преодоления практическую и промышленной криптографии является стандартом де-факто, а также международные организации признаются в качестве официального стандарта. Асимметричная алгоритм шифрования является довольно распространенным типом образом. С другой стороны, RSA-программное обеспечение для свободного обращения, алгоритм RSA ограничена в США защищены несколько патентов. Способ RSA, шифрование / deshifrovyvaniya, также создать цифровую подпись, и процесс проверки используют в качестве процесса.

1.4.1.1 Генерация ключа

Первым этапом генерации ключей любого метода асимметричного шифрования является создание некой пары ключей: открытого ключа, закрытого ключа, а так же обязательная пересылка параметров открытого

ключа всем желающим. Криптоанализ может производиться в совокупности с шифрованием. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д. Метода шифрования RSA данный этап осуществляет путем следующих операций:

Выбор двух больших простых чисел p и q;

Расчет их произведения $n=p \cdot q$;

Выбор произвольного простого большого числа e ($e<n$), такого, что НОД$(e,(p-1)(q-1))=1$, то есть параметр e должно быть взаимно простым с функцией эвклида. Достаточно распространенным алгоритмом шифрования ассиметричного типа является метод.

Методом Евклида решает целый ряд вопросов, связанных с вычислением простых чисел с помощью уравнения [3,4] $e \cdot d+(p-1)(q-1) \cdot y=1$. Неизвестными параметрами в данном случае будут следующие переменные типа d и y. Метод вычисления таких чисел находит целое множество пар типа (d,y), каждая из которых является одним из решений данного уравнения. Два числа типа (e,n) – публикуютсяв открытом доступе и являются открытым ключом.

Число d хранится в строжайшем секрете – это и есть закрытый ключ, который позволит читать все послания, зашифрованные с помощью пары чисел (e,n).

1.4.1.2 Шифрование/расшифровывание

Отправитель разбивает свое сообщение на блоки, равные $k=[\log_2(n)]$ бит, где квадратные скобки обозначают взятие целой части от дробного числа.

Подобный такой блок может быть так же выражен как некое число из представленного диапазона $(0;2^k-1)$. Поэтому для этих чисел (m_i) рассчитывается следующее выражение типа $c_i=((m_i)^e) \bmod n$. Данные блоки c_i и будут этим зашифрованным сообщением, так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных, и их возможно с легкостью передавать по любому открытому каналу, так как операции возведения в степень по модулю простого числа, являются сложной и необратимой математической задачей. Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д. Асимметричная алгоритм шифрования является довольно распространенным типом образом. Проблема Контакт "ограниченная область логарифмов" и сфера поселка под названием очень сложные несколько заказов. Если даже злоумышленник знает наши числа e и n, то по сообщению c_i прочесть исходное сообщение m_i он никогда не сможет, если только будет производить полный перебор m_i [3].

На принимающей же стороне данный процесс дешифрования имеет место, с помощью секретного числа d. Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д. Была доказана теорема Эйлера, как частный случай которой говорит о том, что если простое число n можно представить в виде двух простых больших чисел p и q, то для любого x имеет место следующее равенство $(x^{(p-1)(q-1)})\text{mod } n = 1$. Для процесса дешифрования [4] RSA-сообщений воспользуемся этой формулой.

Возведем обе ее части в степень [9] (-y) : $(x^{(-y)(p-1)(q-1)})\text{mod } n = 1^{(-y)} = 1$.

Теперь умножим обе ее части на x : $(x^{(-y)(p-1)(q-1)+1})\text{mod } n = 1*x = x$.

А теперь вспомним как создавались открытый и закрытый ключи. с помощью алгоритма Евклида подбиралось такое d, что [9] e*d+(p-1)(q-1)*y=1, то есть e*d=(-y)(p-1)(q-1)+1. А следовательно в последнем выражении предыдущего абзаца можем заменить показатель степени на число (e*d). Получаем $(x^{e*d})\text{mod } n = x$. То есть для того чтобы прочесть сообщение $c_i=((m_i)^e)\text{mod } n$ достаточно возвести его в степень d по модулю m:

$$((c_i)^d)\text{mod } n = ((m_i)^{e*d})\text{mod } n = m_i. \qquad (1.4)$$

В самом деле, даже если они действительно любое время в оптимальный алгоритм и достаточно о возведения в степень обработки высокой большой современный для деятельности простых. Криптоанализ может производиться в совокупности с шифрованием [9]. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д. Таким образом, вся зашифрованный текст обычно блочный шифр (быстрый), закодированного сообщения, но с использованием ключа сеанса, но сам ключ использует открытый ключ алгоритма неравенство получателя только зашифрованные сеансы, и помещается в начало файла [4].

1.4.2 Алгоритм Эль-Гамаля

В 1985 году Т.Эль-Гамаль (США) предложил такую схему, которая основана на возведения в степень по модулю большого простого числа P. Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д. Задается большое простое число P и целое число A, $1 < A < P$. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных.

Сообщения представляются целыми числами M из интервала $1 < M < P$.

1.4.2.1 Шифрование сообщений [3]

Протокол передачи такого сообщения M выглядит следующим образом: абоненты уже знают числа A и P и генерируют независимо друг от друга случайные большие числа: Ka, Kb, удовлетворяющих условию:

$$1 < K < P \tag{1.5}$$

получатель вычисляет и передаёт отправителю такое число B, которое определяется следующей последовательностью:

$$B = A^{Kb} \bmod(P) \tag{1.6}$$

отправитель шифрует данное сообщение M и отправляет полученную последовательность получателю:

$$C = M * B^{Ka} \bmod(P) \tag{1.7}$$

получатель расшифровывает полученное им сообщение:

$$D = (A^{Ka})^{-Kb} \bmod(P), \tag{1.8}$$

$$M = C * D \bmod(P) \tag{1.9}$$

В данной системе открытого процесса шифрования та же степень защиты, что для алгоритма RSA с модулем N из 200 знаков, достигается уже при модуле P из 150 знаков. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных. Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д. Это позволяет в 5-7 раз увеличить скорость обработки информации. Однако, в таком варианте открытого процесса шифрования нет подтверждения подлинности сообщений.

1.4.2.2 Подтверждение подлинности отправителя

Для того, чтобы обеспечить при открытом шифровании по модулю простого числа P также и процедуру подтверждения подлинности отправителя Т.ЭльГамаль предложил следующий протокол передачи подписанного сообщения M: абоненты знают числа A и P [3] , отправитель генерирует случайное число и хранит его в секрете: Ka, удовлетворяющее условию:

$$1 < Ka < P \tag{1.20}$$

вычисляет и передаё получателю число B, определяемое [3] последовательностью:

$$B = A^{Ka} \bmod(P) \qquad (1.21)$$

Для сообщения M ($1 < M < P$) выбирает случайное число L ($1 < L < P$), удовлетворяющее условию:

$$(L , P - 1) = 1 \qquad (1.22)$$

вычисляет число: $R = A^{L} \bmod(P)$
решает относительно S

$$M = Ka * R + L * S \bmod(P) \qquad (1.23)$$

передаёт подписанное сообщение: [M, R, S]
получатель проверяет правильность подписи:

$$A\ M = (B^{R}) * (R^{S}) \bmod(P) \qquad (1.24)$$

В этой системе секретным ключом для подписывания сообщений является число X, а открытым ключом для проверки достоверности подписи число B. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных. Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д. Процедура проверки подписи служит также и для проверки правильности расшифровывания, если сообщения шифруются. Достаточно распространенным алгоритмом шифрования ассиметричного типа является метод.

2 Практическое применение криптологии

2.1 Цифровая подпись

При ведении деловой переписки, при заключении контрактов подпись ответственного лица является непременным атрибутом документа, преследующим несколько целей:

– гарантирование истинности письма путем сличения подписи с имеющимся образцом;

– гарантирование авторства документа (с юридической точки зрения).

Выполнение данных требований основывается на следующих свойствах подписи:

– подпись аутентична, то есть с ее помощью получателю документа можно доказать, что она принадлежит подписывающему;

– подпись [7] не подделываема, то есть служит доказательством, что только тот человек, чей автограф стоит на документе, мог подписать данный документ, и никто иной;

– подпись непереносима, то есть является частью документа и поэтому перенести ее на другой документ невозможно;

– документ с подписью является неизменяемым;

– подпись неоспорима;

– любое лицо, владеющее образцом подписи может удостоверится, что документ подписан владельцем подписи;

– развитие современных средств безбумажного документооборота, средств электронных платежей немыслимо без развития средств доказательства подлинности и целостности документа. Таким средством является электронно-цифровая подпись (ЭЦП), которая сохранила основные свойства обычной подписи.

Существует несколько методов [4] построения ЭЦП, а именно:

шифрование электронного документа (ЭД) на основе симметричных алгоритмов. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных. Данная схема предусматривает наличие в системе третьего лица – арбитра, пользующегося доверием обеих сторон. Авторизацией документа в данной схеме является сам факт шифрования ЭД секретным ключом и передача его арбитру. Достаточно распространенным алгоритмом шифрования ассиметричного типа является метод;

– использование ассиметричных алгоритмов шифрования. Фактом подписания документа является шифрование его на секретном ключе отправителя.

Развитием предыдущей идеи стала наиболее распространенная схема ЭЦП – шифрование окончательного результата обработки ЭД хеш-функцией при помощи ассиметричного алгоритма.

Кроме перечисленных, существуют и другие методы построения схем ЭЦП - групповая подпись, неоспариваемая подпись, доверенная подпись и др. Появление этих разновидностей обусловлено разнообразием задач, решаемых с помощью электронных технологий передачи и обработки электронных документов.

2.2 Алгоритм DSA

В 1991 г. в США был опубликован проект федерального стандарта цифровой подписи - DSS (Digital Signature Standard, [DSS91], описывающий систему цифровой подписи DSA (Digital Signature Algorithm). Одним из основных критериев при создании проекта была его патентная чистота.

Предложенный алгоритм, суточные, RSA, а также ряд теорий относительно того, качество и Шнорра Эль-Гамаля представляет собой криптографический система, основанная версия. Его дискретных логарифмов надежного компьютера, основанного на конкретном случае, в почти непреодолимые проблемы. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных. Чтобы решить эту проблему, современный подход к теми же преимуществами и часть навыков решения; Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д. Таким образом, RSA с той же надежностью, как количество бит система 512-1024 ключевой рекомендуется. Схема подписи DSA RSA меньше 320 бит в длину. Асимметричное шифрование Довольно распространенной формой [12].

Публикация получила много критики проекта включены в окончательный поправок права. Асимметричное шифрование Довольно распространенной формой главных аргументов против DSA metod. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных. Odnim Генеральный расчета задачи дискретного логарифмирования, в отличие от плохого понимания и нападения немного сложнее, и это является частным случаем использования системы. Кроме того, стандарты, которые используются для создания цифровой подписи, псевдо-случайных чисел и метод для изготовления этой части алгоритма является криптографическая защита является одним из самых важных.

Функция DSA только ограниченных цифровых подписей, главная цель системы является для шифрования данных. Подписи DSA, RSA и сравнить система ставок, но значительное (10-40) дал ему во время подписания проверки

[4]. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных.

Асимметричная алгоритм шифрования является довольно распространенным типом образом.

2.2.1 Генерация ЭЦП

При генерации ЭЦП используются параметры трех групп:

– общие параметры;

– секретный ключ;

– открытый ключ.

Общие параметры необходимы для функционирования системы в целом. Секретный ключ используется для формирования ЭЦП, а открытый – для проверки ЭЦП. Общими параметрами системы являются простые целые числа p,q,g, удовлетворяющие следующим условиям [7]: p: 2^511<p<2^512

q: простой делитель числа (p-1), который удовлетворяет условию:

2^159<q<2^160

g: так называемый генератор, удовлетворяющий равенству:

g=h^((p-1)/q)mod p >1.

Параметры p,q,g публикуются для всех участников обмена ЭД с ЭЦП.

Достаточно распространенным алгоритмом шифрования ассиметричного типа является метод. Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д. Секретный ключ x случайно выбирается из диапазона [1,q] и держится в секрете. Открытый ключ вычисляется: y=g^x mod p.

Также при описании данной схемы будут использоваться следующие обозначения и дополнительные параметры: m – входное сообщение пользователя для схемы с ЭЦП; k - случайное число, удовлетворяющее условию 0<k<q, хранящееся в секрете и меняющееся от одной подписи к другой; H – хэш-функция, h – хэш-код сообщения.

Процесс генерации ЭЦП состоит из нескольких этапов:

1.Вычисляется хэш-код сообщения m h=H(m)

2.Из диапазона [1,q] случайным образом выбирается значение k и вычисляется r= (g^k mod p) mod q

3. Вычисляется S= (k^-1(h+xr)) mod q, где k^-1 удовлетворяет условию

(k^-1*k) mod q =1

Значения r,s являются ЭЦП сообщения m и передаются вместе с ним по каналам связи.

2.2.2 Проверка ЭЦП

Пусть принято сообщение m1 и его подпись s1,r1.

Проверка ЭЦП происходит следующим образом [4]:

– проверяется [4] выполнение условий 0<r1<q, 0<s1<q, и если хотя бы одно из них нарушено, подпись отвергается.

Вычисляются значения:

w= s1^-1 mod q
u1 = (H(m1)w) mod q
u2 = ((r1/w) mod q
v = ((g^u1y^u2) mod p) mod q
проверяется равенство v = r1

Если последнее равенство выполняется, то подпись принимается. В данном стандарте специфицируется также процедура генерации основных параметров системы и проводится доказательство того, что если v=r1, то [7] m1=m, r1=r, s1=s.

2.3 Алгоритм DES

Принятие стандарта шифрования DES явилось мощным толчком к широкому применению шифрования в коммерческих системах. Введение этого стандарта - отличный пример унификации и стандартизации средств защиты. Примером системного подхода к созданию единой крупномасштабной системы защиты информации является директива Министерства финансов США 1984 года, согласно которой все общественные и частные организации, ведущие дела с правительством США, обязаны внедрить процедуру шифрования DES; крупнейшие банки Citibank,Chase Manhattan Bank, Manufaktures Hannover Trust, Bank of America, Security Pacific Bank также внедрили эту систему.

Министерство энергетики США располагает более чем 30 действующими сетями, в которых используется алгоритм DES, Министерство юстиции устанавливает 20000 радиоустройств, располагающих средствами защиты на базе DES. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных. Достаточно распространенным алгоритмом шифрования ассиметричного типа является метод. Стандартизация в последнее время приобретает международный характер, подтверждение тому - международный стандарт 1987 года ISO 8372, разработанный на основе криптоалгоритма DES.

В качестве стандартной аппаратуры шифрования можно назвать устройство Cidex-HX, базирующееся на алгоритме DES; скорость шифрования - от 56 Кбит/с до 7 Мбит/с. Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д. Серийно выпускается автономный шифровальный блок DES 2000, в нем также используется процедура шифрования DES; скорость шифрования - от 38,4 Кбит/с до 110 Кбит/с. В

различных секторах коммерческой деятельности используется процессор шифрования/дешифрования данных FACOM 2151A на основе алгоритма DES; скорость - от 2,4 Кбит/с до 19,2 Кбит/с. С распространением персональных компьютеров наиболее эффективными для них стали программные средства защиты. Достаточно распространенным алгоритмом шифрования ассиметричного типа является метод. Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д. Так, разработан пакет программ для шифрования/дешифрования информации CTA (Computer Intelligence Access), реализующий алгоритм DES. Этот же алгоритм использован в пакете SecretDisk (C F Systems) для исключения несанкционированного доступа к дискам.

Таким образом, алгоритм DES представляет собой основной механизм, применявшийся частными и государственными учреждениями США для защиты информации. В то же время Агентство национальной безопасности, выступающее как эксперт по криптографическим алгоритмам, разрабатывает новые алгоритмы шифрования данных для массового использования. Достаточно распространенным алгоритмом шифрования ассиметричного типа является метод. В 1987 году Национальное бюро стандартов после обсуждения подтвердило действие DES; его пересмотр намечалось провести не позднее января 1992 года, и на сегодняшний день действие DES ограничивается исключительно коммерческими системами [7].

2.4 Алгоритм MD5

Алгоритм MD5 был создан в 1991 году профессором Массачусетского Технологического Института (MIT, Massachusetts Institute of Technlogy) Рональдом Райвестом (Ronald Rivest) в целях создания цифровых подписей. Он предназначен для использования на 32-битных машинах и является более безопасным, нежели алгоритм MD4, который был сломан. MD5 - односторонняя хэш-функция, то есть, зная лишь результат преобразования, невозможно восстановить исходную информацию. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных. Достаточно распространенным алгоритмом шифрования ассиметричного типа является метод.

Таким образом, алгоритм MD5 преобразует исходную информацию в число фиксированной длины, называемое "дайджестом сообщения" (message digest).

Использование MD5 позволяет сравнить дайджест сообщения с опубликованным, чтобы убедиться, что данное сообщение полностью совпадает с оригинальным, то есть, не было повреждено или изменено. Данная процедура сравнения называется "проверка хэша" (hashcheck).

Проверочная сумма MD5 (MD5 checksum) пустого сообщения:

d41d8cd98f00b204e9800998ecf8427e

Цифровая подпись - это базовый элемент в криптографии играет решающую роль в аутентификации и авторизации. Сигнатура проблемы - выявление игрока с передаваемых данных. Криптоанализ может производиться в совокупности с шифрованием. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д. Процесс подписи заключается в преобразовании исходного сообщения и некоторую секретную информацию для надлежащего tsiforvuyu подписи. Довольно распространенной формой асимметричного метода шифрования.

Как обычно подписи, цифровая подпись, чтобы убедиться, что сообщение было отправлено с тем, кто называет себя отправитель. Цифровые подписи особенно важны в области электронной коммерции и являются ключевыми элементами в различных схем для проверки подлинности.

Цифровая подпись должна быть некоторое сопротивление, а именно целесообразность его использования вызывает сомнения. Алгоритм MD5 является одним из многих алгоритмов, направленных на обеспечение необходимого уровня безопасности. Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных.

Контрольная сумма MD5 часто используется для проверки целостности загруженных файлов. Сравнивая сумма MD5 скачал файл с опубликованными значениями, пользователь может убедиться, что изображение вы получили такое же, как и оригинал, не содержит вирусов.

Алгоритм MD5 широко использовался и используется, и изначально считалось, что он абсолютно криптоустойчив. Однако, в 1994 году была открыта уязвимость, которая поставила под вопрос дальнейшее использование алгоритма. Было показано, что возможно создавать пары сообщений, имеющие одну и ту же проверочную сумму. Достаточно распространенным алгоритмом шифрования ассиметричного типа является метод.

Под коллизией (collision) хэш-функции понимается получение одного и того же значения для разных сообщений при идентичном начальном буфере. Если же начальные буферы различаются, то совпадение выходных значений (как для разных сообщений, так и для одинаковых) называется псевдоколлизией (pseudo-collision).

В 1993 [4] году Bert den Boer и Antoon Bosselaers показали, как можно обнаружить псевдоколлизии в алгоритме MD5. Matt Robshaw так прокомментировал эту атаку:

"На самом деле псевдоколлизия возникает при инициализации буфера из четырех слов при запуске алгоритма MD5 двумя разными значениями. Эти значения различаются только своими старшими разрядами в каждом слове. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных. Для обоих буферов используется одно и то же сообщение, при этом получается одинаковый дайджест сообщения. Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д.

Если бы можно было выбрать одно и то же стартовое значение для буфера (необязательно то, что применяется в алгоритме), и затем выбрать два разных сообщения, возможно, различающихся лишь несколькими битами в каком-нибудь слове, таким образом, что получался бы один и тот же дайджест, то это было бы намного более серьезной уязвимостью." Достаточно распространенным алгоритмом шифрования ассиметричного типа является метод.

2.4.1 Процесс шифрования

На вход алгоритма поступает входной поток данных, хеш которого необходимо найти. Длина сообщения может быть любой (в том числе нулевой). Запишем длину сообщения в L. Это число целое и неотрицательное. Кратность каким-либо числам необязательна. После поступления данных идёт процесс подготовки потока к вычислениям [10].

Шаг 1. Выравнивание потока.

Сначала дописывают единичный бит в конец потока (байт 0x80), затем необходимое число нулевых бит. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных. Входные данные выравниваются так, чтобы их новый размер L' был сравним с 448 по модулю [10] 512 ($L' = 512 \times N + 448$). Выравнивание происходит, даже если длина уже сравнима с 448.

Шаг 2. Добавление длины сообщения.

В оставшиеся 64 бита дописывают 64-битное представление длины данных (количество бит в сообщении) до выравнивания. Сначала записывают младшие 4 байта. Если длина превосходит $2^{64} - 1$, то дописывают только младшие биты. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных. После этого длина потока станет кратной 512. Вычисления будут основываться на представлении этого потока данных в виде массива слов по 512 бит.

Шаг 3. Инициализация буфера.

Для вычислений инициализируются 4 переменных размером по 32 бита и задаются начальные значения шестнадцатеричными числами (шестнадцатеричное представление, сначала младший байт) [10]:

A = 01 23 45 67;

B = 89 AB CD EF;

C = FE DC BA 98;

D = 76 54 32 10.

В этих переменных будут храниться результаты промежуточных вычислений. Начальное состояние ABCD называется инициализирующим вектором.

Определим ещё функции и константы, которые нам понадобятся для вычислений.

Потребуются 4 функции (1)-(4) для четырёх раундов. Введём функции от трёх параметров - слов, результатом также будет слово.

1 раунд $FunF(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge Z)$ (2.1)

2 раунд $FunG(X,Y,Z) = (X \wedge Z) \vee (\neg Z \wedge Y)$ (2.2)

3 раунд $FunH(X,Y,Z) = X \oplus Y \oplus Z$ (2.3)

4 раунд $FunI(X,Y,Z) = Y \oplus (\neg Z \vee X)$ (2.4)

Определим таблицу констант $T[1..64]$ - 64-элементная таблица данных, построенная следующим образом [10]:

$$T[i] = int(4294967296 \cdot |\sin(i)|), \text{ где } 4294967296 = 2^{32} \quad (2.5)$$

Выровненные данные разбиваются на блоки (слова) по 32 бита, и каждый блок проходит 4 раунда из 16 операторов. Все операторы однотипны и имеют вид [abcd k s i], определяемый по формуле (6):

$$a = b + ((a + Fun(b,c,d) + X[k] + T[i]) <<< s), \quad (2.6)$$

где X – блок данных;

$X[k] = M[n \cdot 16 + k]$, где k – номер 32-битного слова из n-го 512 битного блока сообщения;

s – циклический сдвиг влево на s бит полученного 32-битного аргумента.

Шаг 4. Вычисление в цикле.

Заносим в блок данных элемент n из массива. Сохраняются значения A, B, C и D, оставшиеся после операций над предыдущими блоками (или их начальные значения, если блок первый). Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных.

```
/*[abcd k s i] a = b + ((a + F(b,c,d) + X[k] + T[i]) <<< s). */
[ABCD  0 7  1][DABC  1 12  2][CDAB  2 17  3][BCDA  3 22  4]
[ABCD  4 7  5][DABC  5 12  6][CDAB  6 17  7][BCDA  7 22  8]
[ABCD  8 7  9][DABC  9 12 10][CDAB 10 17 11][BCDA 11 22 12]
[ABCD 12 7 13][DABC 13 12 14][CDAB 14 17 15][BCDA 15 22 16]

/*[abcd k s i] a = b + ((a + G(b,c,d) + X[k] + T[i]) <<< s). */
[ABCD  1 5 17][DABC  6 9 18][CDAB 11 14 19][BCDA  0 20 20]
[ABCD  5 5 21][DABC 10 9 22][CDAB 15 14 23][BCDA  4 20 24]
[ABCD  9 5 25][DABC 14 9 26][CDAB  3 14 27][BCDA  8 20 28]
[ABCD 13 5 29][DABC  2 9 30][CDAB  7 14 31][BCDA 12 20 32]

/*[abcd k s i] a = b + ((a + H(b,c,d) + X[k] + T[i]) <<< s). */
[ABCD  5 4 33][DABC  8 11 34][CDAB 11 16 35][BCDA 14 23 36]
[ABCD  1 4 37][DABC  4 11 38][CDAB  7 16 39][BCDA 10 23 40]
[ABCD 13 4 41][DABC  0 11 42][CDAB  3 16 43][BCDA  6 23 44]
[ABCD  9 4 45][DABC 12 11 46][CDAB 15 16 47][BCDA  2 23 48]

/*[abcd k s i] a = b + ((a + I(b,c,d) + X[k] + T[i]) <<< s). */
[ABCD  0 6 49][DABC  7 10 50][CDAB 14 15 51][BCDA  5 21 52]
[ABCD 12 6 53][DABC  3 10 54][CDAB 10 15 55][BCDA  1 21 56]
[ABCD  8 6 57][DABC 15 10 58][CDAB  6 15 59][BCDA 13 21 60]
[ABCD  4 6 61][DABC 11 10 62][CDAB  2 15 63][BCDA  9 21 64]
```

Раунд 1

Раунд 2

Раунд 3

Раунд 4

Рисунок 2.1 – Четыре раунда преобразований в алгоритме MD5

Суммируем с результатом предыдущего цикла:

A = AA + A

B = BB + B

C = CC + C

D = DD + D

После окончания цикла необходимо проверить, есть ли ещё блоки для вычислений. Если да, то изменяем номер элемента массива (n++) и переходим в начало цикла /2/.

Шаг 5. Результат вычислений с использованием готового ПО

Результат вычислений находится в буфере ABCD, это и есть хеш. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных. Если выводить побайтово начиная с младшего байта A и закончив старшим байтом D, то мы получим MD5 хеш [10].

2.5 Алгоритм SHA-1

Алгоритм SHA-1 (Secure Hash Algorithm) был разработан Национальным Агентством Безопасности (NSA, National Security Agency) и опубликован Национальным Институтом Стандартов и Технологии (NIST, National Institute of Standards and Technology). Этот алгоритм сопоставляет сообщению с максимальной длиной 64 бита дайджест длиной 160 бит. Достаточно распространенным алгоритмом шифрования ассиметричного типа является метод. Первоначальная спецификация алгоритма была опубликована в 1993 году под названием Secure Hash Standard, FIPS PUBS 180 (Federal Information Processing Standards Publications). Данную версию теперь

часто называют SHA-0. NSA отказалась от нее вскоре после публикации и заменила улучшенной версией, опубликованной в 1995 году в FIPS PUBS 180-1 и обычно называемой SHA-1.

Согласно заявлениям NSA, это было сделано в целях исправления ошибки в оригинальном алгоритме, которая уменьшала его криптографическую устойчивость. Достаточно распространенным алгоритмом шифрования ассиметричного типа является метод.

Однако, NSA не сообщила никакой дополнительной информации. Много спустя, на конференции Crypto в 1998 году два французских исследователя (F. Chabaud и A. Joux) представили атаку на алгоритм SHA-0, которая не работала на алгоритме SHA-1. Возможно, это и была ошибка, открытая NSA. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных. Алгоритм SHA-1 был тщательно изучен криптографическим сообществом и пока не было найдено никаких уязвимостей. Таким образом, он считается вполне безопасным.

NIST также опубликовал три дополнительных вариации алгоритма SHA с более длинными дигетами. Их названия соответствуют длине дигета: SHA-256, SHA-384 и SHA-512. Они были впервые опубликованы в 2001 году в черновом варианте FIPS PUBS 180-2. Официальная версия FIPS PUBS 180-2, которая также включает SHA-1, была выпущена в качестве официального стандарта в 2002 году. Новые хэш-функции еще не были изучены с той же тщательностью, что SHA-1, поэтому их криптоустойчивость пока не подтверждена. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных.

Алгоритм MD5 является последователем MD4 с улучшенным побитовым хешированием, дополнительным раундом и улучшенным "лавинным эффектом" (avalanche effect).

Алгоритм SHA также происходит от MD4 и отличается от последнего расширенной трансформацией, дополнительным раундом и улучшенным "лавинным эффектом". Криптоанализ может производиться в совокупности с шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д.

Несмотря на то, что до сих пор не был найден способ атаки MD5 на практике, тот факт, что такая атака возможна, внушает опасения. SHA-1 с более длинным дайджестом и устойчивостью к подобным атакам выглядит предпочтительнее.

Преимущество MD5 перед SHA-1 - производительность

Производительность измерялась в мегабайтах в секунду.

Различия алгоритмов MD4 и MD5:

– в алгоритме MD5 был добавлен один раунд в обработке блоками по 16 слов;

– на каждом шаге прибавляется уникальная добавочная константа;

– изменена функция G для большей симметричности;

– на каждом шаге прибавляется результат предыдущего шага;

– порядок работы с входными словами в раундах 2 и 3 был изменен для большего различия;

– были оптимизированы сдвиги в каждом раунде.

2.6 Повышение криптостойкости DES

Чтобы увеличивать криптостойкость DES появляются несколько вариантов: double DES, triple DES, DESX, G-DES.

– Методы 2DES и 3DES основаны на DES, но увеличивают длину ключей и поэтому увеличивается криптостойкость.

– Схема 3DES имеет вид DES)), где k1,k2,k3 ключи для каждого шифра DES. Это вариант известен как в EEE так как три DES операции являются шифрованием. Существует 3 типа алгоритма 3DES:

– DES-EEE3: Шифруется три раза с 3 разными ключами.

– DES-EDE3: 3DES операции шифровка-расшифровка-шифровка с 3 разными ключами.

– DES-EEE2 и DES-EDE2: Как и предыдущие, за исключением того, что первая и третья операции используют одинаковый ключ. Достаточно распространенным алгоритмом шифрования ассиметричного типа является метод. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных.

Самый популярный тип при использовании 3DES — это DES-EDE3, для него алгоритм выглядит так:

Зашифрование: $C = E_{k3} E_{k2}^{-1}(E_{k1}(P))$.

Расшифрование: $P = E_{k1}^{-1}(E_{k2}(E_{k3}^{-1}(C)))$

При выполнении алгоритма 3DES ключи могут выбрать так:

– k1,k2,k3 независимы.

– k1,k2 независимы, а k1 = k3

– k1 = k2 = k3.

Метод DESX создан Рональдом Ривестом и формально продемонстрирована Killian и Rogaway. Этод метод — усиленный вариант DES, поддерживаемый инструментарием RSA Security. DESX отличается от DES тем, что каждый бит входного открытого текста DESX логически суммируется по модулю 2 с 64 битами дополнительного ключа, а затем шифруется по алгоритму DES. Каждый бит результата также логически суммируется по модулю 2 с другими 64 битами ключа. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных. Криптоанализ может производиться в совокупности с

шифрованием. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д. Достаточно распространенным алгоритмом шифрования ассиметричного типа является метод. Главной причиной использования DESX является простой в вычислительном смысле способ значительного повысить стойкость DES к атакам полного перебора ключа.

Метод G-DES разработан Schaumuller-Bichl для повышения производительности DES на основе увеличения размером шфрованного блока. Заявлялось, что G-DES защищен так же как и DES. Однако, Biham и Shamir показали, что G-DES с рекомендуемыми параметрами легко взламывается, а при любых изменениях параметров шифр становится ещё менее защищен чем DES.

Другой вариант DES использует независимых суб-ключей. Различные алгоритм DES, который основан на 56-битной секретного ключа, который пользователь получает алгоритм DES шестнадцать 48-битных суб-ключей для каждого из 16 раундов, эта реализация использует 768 битный ключ вместо 16 дочерних 48-битные ключи, ключевое слово Graphics алгоритм DES. Довольно распространенной формой асимметричного метода шифрования. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных.

Хотя очевидно, что использование независимых суб-ключей усложнить всего ключа к поиску, но устойчивость к различным атакам линейного криптоанализа не намного сопротивление превышает нормальную DES. Рейтинг Бихам для дифференциального криптоанализа DES с независимыми ключами требует двух выбранных открытых текстов, в то время как линейный криптоанализ требует 2 известных открытых текстов.

2.7 Зависимость криптостойкости от ключа

Любую секретную информацию можно получить путем перебора всех возможных ключей, поэтому проведем оценку возможности подбора ключей. Проблема поиска ключей симметричной криптосистемы путем перебора всех возможных ключей относится к классу задач, допускающих распараллеливание. Таким образом, прогресс в решении подобных задач возможен за счет:

- увеличения производительности отдельного процессора;
- увеличения количества процессоров в системе.

Попробуйте проанализировать пределы этих двух движений. По нашим оценкам, максимальная эффективность вычислительных устройств связаны с определением максимальной эффективности, на основе физических законов нашего мира. Максимальная скорость передачи данных в нашей вселенной - скорость света, максимальная записи информации плотность - бит на атом. Довольно распространенной формой асимметричного метода шифрования.

Высокоскоростная передача данных не может быть основано на законах физики, запись высокой плотности не возможно в соответствии с неопределенности Гейзенберга. Криптоанализ может производиться в совокупности с шифрованием. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д.

Предположим, что размер процессора равен размеру атома. Тогда в наших обозначениях быстродействие гипотетического процессора выразится формулой $F = Vc/Ra = 3 * 1018$операций в секунду, где $Vc = 3 * 10$ 8 м/с скорость света в вакууме, а $Ra = 10\text{-}10$ м - размеры атомов. Столько раз за 1 секунду свет пройдет размеры атома. Поскольку период обращения Земли вокруг Солнца составляет 365,2564 суток или 31 558 153 секунд, то за один год такой процессор выполнит $94\,674\,459 * 1018 \approx 1026$ операций. Более быстрый процессор в нашей вселенной невозможен в принципе. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных. За 100 лет непрерывной работы гипотетический процессор совершит приблизительно 10^{28} операций. При условии, что за один такт своей работы он проверяет один ключ, а расшифровка сообщения на найденном ключе происходит мгновенно, то он сможет перебрать 10^{28} ключей, т.е. длина ключа составит всего лишь 93 бита!

Очевидно, что чем больше высокая скорость, то можно только за счет увеличения количества процессоров в системе. Другие способы увеличить вычислительную мощность нет. Довольно распространенной формой асимметричного метода шифрования. Анализ пороговых значений, указанных во второй тенденции, следует отметить, что тоже является увеличение числа процессоров в системе имеет свои пределы.

Для нашей планеты естественным пределом является площадь земной поверхности. Если выразить поверхность земного шара (считая океаны, пустыни, Арктику с Антарктикой) в квадратных миллиметрах, и на каждый миллиметр поместить по миллиону таких процессоров, то в год мощность такого вычислительного устройства составит $5.1 * 10^{52}$ операций, что эквивалентно длине в 175-176 бит. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных. Если исходить из предположения, что стойкость шифра должна составлять 100 лет, то за указанный период такая система сможет перебрать $5 *10^{54}$ ключей, что составит 181-182 бита. И это притом, что никакие вычислительные ресурсы процессоров не тратятся на согласование их взаимной работы в системе, на решение задачи дешифрования и т.д.

Из проведенного исследования можно сделать вывод, что для обеспечения надежности достаточно использовать алгоритмы с длиной ключа не менее 64 битов, а применять и разрабатывать алгоритмы с длиной ключа более 128 бит экономически не выгодно. Однако, как правило, для генерации

ключа используется пароль, который в свою очередь часто содержит лишь символы латинского алфавита. Достаточно распространенным алгоритмом шифрования ассиметричного типа является метод. Криптоанализ может производиться в совокупности с шифрованием. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных. Существенно расширяют функционально и другие функции, например, такие как конвертации (текст) по определенным стандартам или использование булевой алгебры и т.д. В таком случае для обеспечения необходимой защиты требуется использовать пароль не короче 12 символов, что соответствует 56-битному ключу. 16-символьный пароль соответствует 75-битному ключу и гарантирует достаточную защиту от прямой атаки.

2.8 Повышение криптостойкости с помощью хеш-функции

Хеш-функция—преобразование текста произвольной длинны в текст фиксированной длинны.

$$H=hash(P),$$

где P—пароль (открытый текст), длинна P от 0 до бесконечности;

H—хеш-значение (хешированный текст), длинна H=N бит (при условии что функция hash возвращает хеш-значение длинной N бит).

Хэш-функция, которая используется алгоритм шифрования. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных. Пользователь в виде фиксированной длины пароль (ключ) хэш-значения любой функции длина хэш преобразует ввода пароля, то функция. (Циклы) в каждой итерации будет увеличиваться криптографический алгоритм.

Ключи находятся по следующему алгоритму:

K1=hash(PASSWORD)

K2=f(K1)

Kn=f(Kn-1)

Функция f—функция преобразования ключа.

Общую схему итерационного алгоритма можно представить следующим образом представленным на рисунке 2.2.

Если знать K1, то возможно вычислить все остальные Ki, i=2..n.

Узнайте пароль, но хэш-функция, не зная значение злоумышленника (хакера) может расшифровать текст. Значения грубой 2S (биты значения S-хэш) необходимы для грубой силы подбора пароля. Хэш функция возвращает 64-битный должны ценить грубой 264-1019 = значение 1.84467441G. компьютер вторжений 1000000 паролей в секунду итерации.

После того, как выбор значений хэш будет максимум 213 503 982 дней.

Если вы усреднять два-264/2 = 232 = 4294967296 значения уменьшается число случаев на «атаки рождения" используется. Эти значения будут обанкротиться всего 1,19 часа.

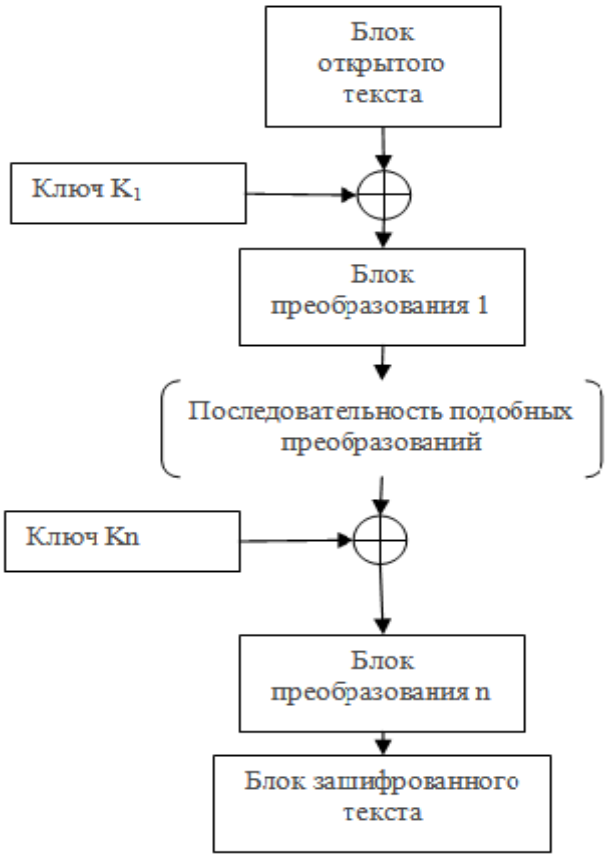

Рисунок 2.2 – Обобщенная схема итерационных криптоалгоритмов

2.9 Метод «изменение процедуры генерации ключей»

Для пароля «Password» были получены следующие хеш-значения:

Таблица 2.2 – Хеш-значения

Хеш-функция	Хеш-значение
MD2	9dc7dd5f9b5f681a133e64c0089330e7
MD4	f15abd57801840f3348ddccafb677f6a
MD5	dc647eb65e6711e155375218212b3964
SHA-1	8be3c943b1609fffbfc51aad666d0a04adf83c9d
SHA-224	a1308b7983fef7def9ffd06bed7ff767fa4216baf9d9d911af1e7e2e
SHA-256	e7cf3ef4f17c3999a94f2c6f612e8a888e5b1026878e4e 19398b23bd38ec221a

SHA-384	d3a1ad34a5f0d265d8c0441d85532a95d02fcca0450646c2 1f1585cfc521843cd423d02f53e9205390706cc15f3a06fe
SHA-512	e6c83b282aeb2e022844595721cc00bbda47cb24537c1779f9 bb84f04039e1676e6ba8573e588da1052510e3aa0a32a9e5587 9ae22b0c2d62136fc0a3e85f8bb
Tiger-128	78db95bcce175ab632095962774965d1
Tiger-160	78db95bcce175ab632095962774965d1151e1f17
Tiger-192	78db95bcce175ab632095962774965d1151e1f1727c63f3d
Whirlpool	fd07ba63996cdcfa6130ee82acec65da7487f51564bb7c6ead 6dabc6b9e8eac974e5d852edc545804ae68fa46fc59d4789acab5 0bbc22b26fb24412f8dc11cde
DES(Unix)	Q37hnXsCsGjCU
CRC-32	9abfd710

Алгоритмы типа CRC-32, DES(Unix) использовать не желательно, т.к. они генерируют очень короткий хеш. А алгоритмы серии Tiger генерируют одинаковые хеши в начале последовательности. Об алгоритмах MD2 и MD4 были заявления о взломе. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных. Из претендентов на реальное использование можно рассматривать только алгоритмы серии SHA, MD5 и Whirlpool.

Наиболее оптимально использовать SHA-512 и Whirlpool.

Предположим, что алгоритм шифрования требует ключ длины 64 бита. Ключ Ks вычисленный по алгоритму SHA-512 занимает 512 бит. Разбиваем ключ на 8 подключей равной длинны. Обозначим их как Ks1, Ks2, вЂ¦, Ks8.

Таблица 2.3 – Значения ключей

Ks1	Ks2	Ks3	Ks4
e6c83b282aeb2e02	2844595721cc00bb	da47cb24537c1779	f9bb84f04039e167
Ks5	Ks6	Ks7	Ks8
6e6ba8573e588da1	052510e3aa0a32a9	e55879ae22b0c2d6	2136fc0a3e85f8bb

Аналогично поступим и с ключом Kw (хеш Whirlpool). Получим 16 подключей Ks1 —Ks8 и Kw1 —Kw8.

Исходя из предположения что в алгоритме рассмотренном на Рисунке 2.1 n=8, возможно итерационную последовательность ключей Ki (i=1..8) заменить независимыми (не итерационными) ключами Kxi (i=1..8).

Тогда злоумышленнику вместо 264 или 232 вариантов придется вычислять 2 хеш-функции, сложность каждой из которых 2512. И того ему придется перебрать:

1) методом «грубой силы»—1.8 Г— 10308 вариантов.
2) атакой «дней рождений»— 1.34 Г— 10154 вариантов.

Таблица 2.4 – Независимые ключи

Ключ	Значение
Kx1	Ks1 XOR Kw5
Kx2	Ks3 XOR Kw2
Kx3	Ks5 XOR Kw3
Kx4	Ks8 XOR Kw1
Kx5	Ks4 XOR Kw7
Kx6	Ks7 XOR Kw4
Kx7	Ks6 XOR Kw8
Kx8	Ks2 XOR Kw6

2.10 Метод «изменения хеш-функции»

Различие в пользователе шифрования к началу пользователя (кто знает пароль) неисправностей (пароль не знаю) начинается, как только пользователь с нужными процесса расшифровки паролей. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных. Нападающий тот же процесс проходит огромное количество времени, чтобы попытаться угадать пароль.

Из этой предпосылки, криптографические алгоритмы могут быть увеличены. Реализация угадывания паролей, криптографический алгоритм уменьшает анализ хеш-функция должна быть медленным. Асимметричная алгоритм шифрования является довольно распространенным типом образом.

Ключ для создания хэш-функции работает на 1 мс. Выберите назвать это функция защиты паролем 1000 раз в секунду при использовании каплю 106 до 1000 секунд (примерно 16 минут) ценностей.

MS 100 хэш-функция должна быть изменена так, что она функционирует. Мы функция, вызываемая slow_hash. 100000 секунд, или 27 часов в slow_hash при использовании хеш функции.

Хеширования пользователе практически не заметно замедляется и взлома займет больше времени, чтобы подписаться.

3 Экспериментальная часть

3.1 Повышение криптостойкости алгоритмов на основе хеш-функции MD5

Пользователь MD5 идентификации счета хеш, и шифрование пароля используется в основном как средство установления сети передачи данных. Например, популярная технология равный-равному, компьютерной сети и хэш-код сверстники (пиров) обмен информацией имеет уникальный "отпечаток пальца" выступает в качестве (соревноваться, торрент-детектив, и т.д.), неотъемлемой частью есть и другие текущая информация. Или, например, сервер электронной почты Пароли MD5 зашифрованные. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных. Асимметричная алгоритм шифрования является довольно распространенным типом образом.

Вход принимает входной поток данных, вы хотите найти алгоритм хэширования. Длина сообщения (в том числе нуль) может быть любой. Это число является целым числом, и мы написали неотрицательное Л.-длинный коридор. Сложение, любое количество народу. После ввода информации во время подготовки вычисления расхода.

Сначала дописывают единичный бит в конец потока (байт 0x80), затем необходимое число нулевых бит. Входные данные выравниваются так, чтобы их новый размер L' был сравним с 448 по модулю 512 (L’ = 512 × N + 448). Выравнивание происходит, даже если длина уже сравнима с 448.

В оставшиеся 64 бита дописывают 64-битное представление длины данных (количество бит в сообщении) до выравнивания. Сначала записывают младшие 4 байта. Если длина превосходит 264 − 1, то дописывают только младшие биты. После этого длина потока станет кратной 512. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных. Вычисления будут основываться на представлении этого потока данных в виде массива

Для вычислений инициализируются 4 переменных размером по 32 бита и задаются начальные значения шестнадцатеричными числами (шестнадцатеричное представление, сначала младший байт):

A = 01 23 45 67;
B = 89 AB CD EF;
C = FE DC BA 98;
D = 76 54 32 10.

В этих переменных будут храниться результаты промежуточных вычислений. Начальное состояние ABCD называется инициализирующим вектором.

Определим ещё функции и константы, которые нам понадобятся для вычислений.

Потребуются 4 функции для четырёх раундов. Введём функции от трёх параметров — слов, результатом также будет слово.

1 раунд $FunF(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge Z)$.
2 раунд $FunG(X,Y,Z) = (X \wedge Z) \vee (\neg Z \wedge Y)$.
3 раунд $FunH(X,Y,Z) = X \oplus Y \oplus Z$.
4 раунд $FunI(X,Y,Z) = Y \oplus (\neg Z \vee X)$.

Определим таблицу констант $T[1..64]$ — 64-элементная таблица данных, построенная следующим образом: $T[i] = int(4294967296 \cdot |sin(i)|)$, где 4294967296 = 2^{32}.

Выровненные данные разбиваются на блоки (слова) по 32 бита, и каждый блок проходит 4 раунда из 16 операторов. Все операторы однотипны и имеют вид: [abcd k s i], определяемый как $a = b + ((a + Fun(b,c,d) + X[k] + T[i]) <<< s)$, где X — блок данных. $X[k] = M[n \cdot 16 + k]$, где k — номер 32-битного слова из n-го 512-битного блока сообщения, и s — циклический сдвиг влево на s бит полученного 32-битного аргумента. Достаточно распространенным алгоритмом шифрования ассиметричного типа является метод.

Заносим в блок данных элемент n из массива. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных. Сохраняются значения A, B, C и D, оставшиеся после операций над предыдущими блоками (или их начальные значения, если блок первый).

AA = A
BB = B
CC = C
DD = D

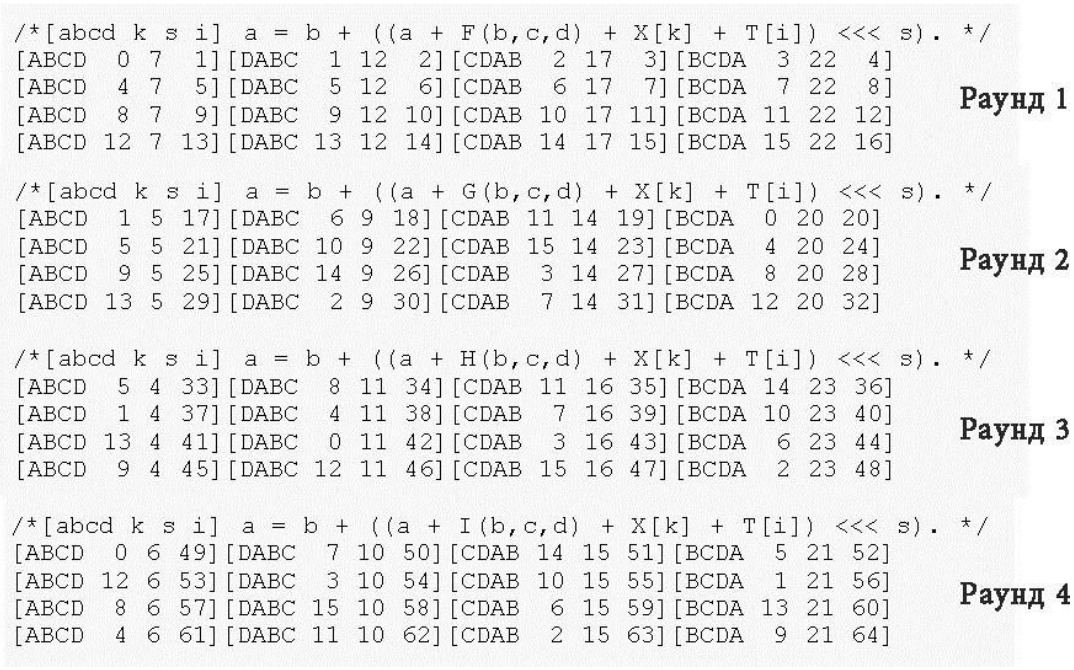

Рисунок 3.1 – Четыре раунда преобразований в алгоритме MD5

Суммируем с результатом предыдущего цикла:

A = AA + A

B = BB + B

C = CC + C

D = DD + D

Цикл более блок вычисления после проверки. Если это так, то количество элементов массива (N) изменение, позвоните в первую очередь. Асимметричная алгоритм шифрования является довольно распространенным типом образом.

Урегулирование это буфер ABCD и хеш. Выходной байт LSB первый байт и старший байт D будет завершена, мы должны получить MD5 хеш. Так что читайте криптографическую является единственным знание является ключом, который может позволить для преобразования данных.

MD5 генератор MD5 алгоритм для шифрования текстовых данных в соответствии с программой очень проста.

Порядок действий при шифровании показан на рисунке 3.2.

С клавиатуры компьютера вводим текстовое сообщение любой длины в поле Исходная строка. Нажимаем Генерировать MD5 и получаем хэш-код введенной текстовой информации (рисунок 3.2).

На данный момент существуют несколько видов «взлома» хешей MD5 — подбора сообщения с заданным хешем:

– Перебор по словарю;
– Brute-force;
– RainbowCrack.

В данном исследовании рассматриваются все три способа взлома кода MD5 при помощи ПО PasswordsPro v 2.5.5.0 с использованием Rainbow tables (радужные таблицы).

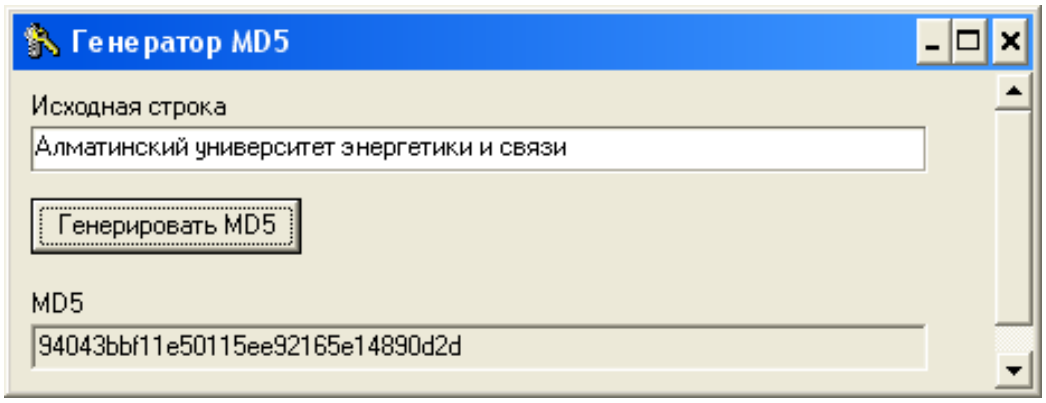

Рисунок 3.2 – Сгенерированный MD5-шифр сообщения

Интерфейс программы представляет собой диалоговое окно (рисунок 3.3) с панелью инструментов, рабочей зоной и строкой поиска.

Программа поддерживает огроманое количество видов хешей и несколько типов атак.

Предварительная атака – это быстрая проверка хешей пользователей на простые пароли типа "123", "qwerty", "99999" и другие, а также на пароли, ранее найденные программой.

Атака полным перебором – это полный перебор всех возможных паролей в каком-либо диапазоне, к примеру – "aaaaaa"..."zzzzzz".

Атака по маске – эта атака используется, если известна какая-либо информация о пароле. Для использования атаки в ее настройках необходимо указать маску для каждого символа в пароле, который требуется восстановить.

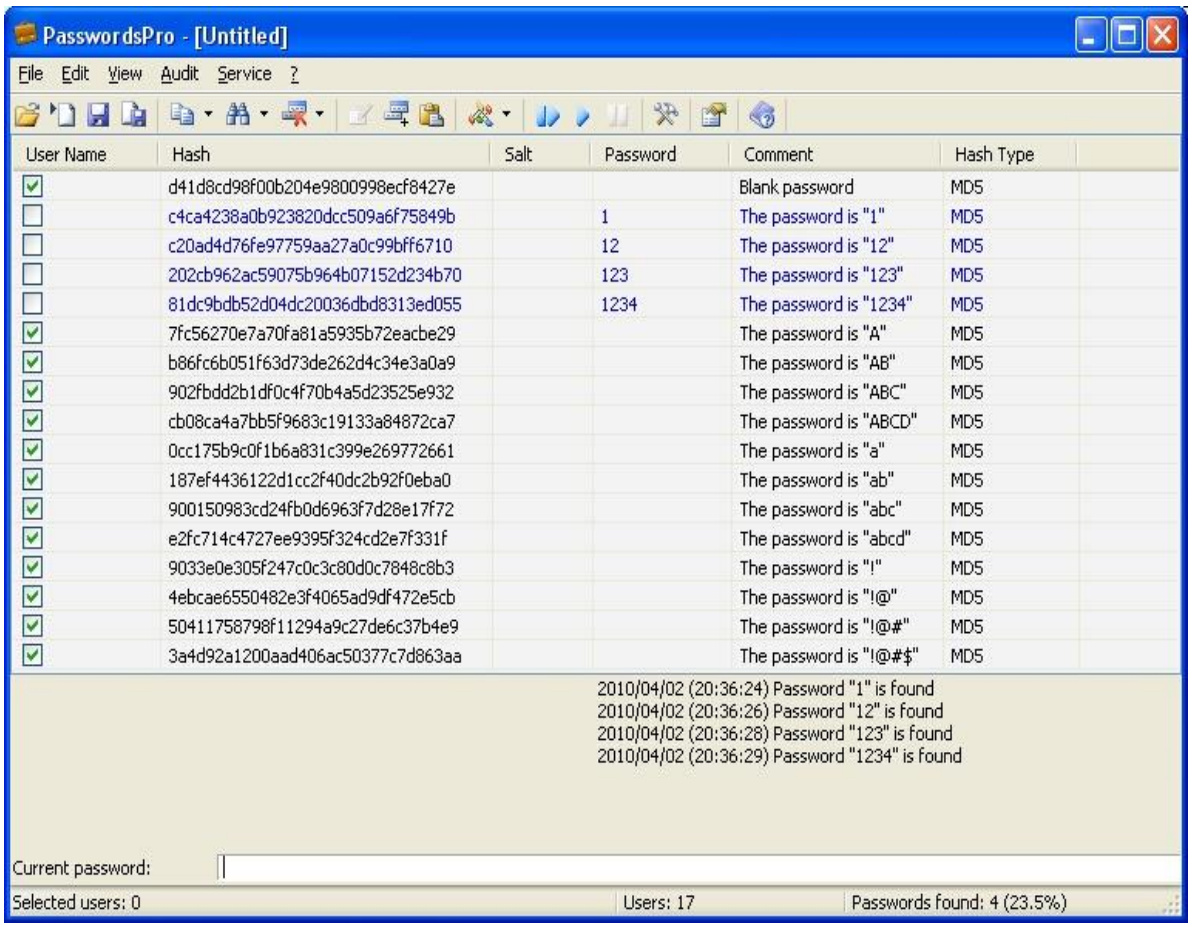

Рисунок 3.3 – Интерфейс программы PasswordsPro

Простая атака по словарям – в этой атаке происходит простая проверка хэшей на пароли из словарей.

Комбинированная атака по словарям – в этой атаке пароли формируются из нескольких слов, взятых из разных словарей, что позволяет восстанавливать сложные пароли вида "superadmin", "admin*admin" и др.

Гибридная атака по словарям – эта атака позволяет изменять пароли из словарей и проверять их в качестве паролей пользователей.

Атака по Rainbow-таблицам – эта атака использует поиск пароля по предварительно рассчитанным Rainbow-таблицам.

3.1.1 Криптоанализ MD5 шифра.

При помощи Генератора MD5 создаем 10 простых хэш-кодов для цифр от 0 до 9. Сохраняем их в текстовом файле (primer.txt) как показано на рисунке 3.4.

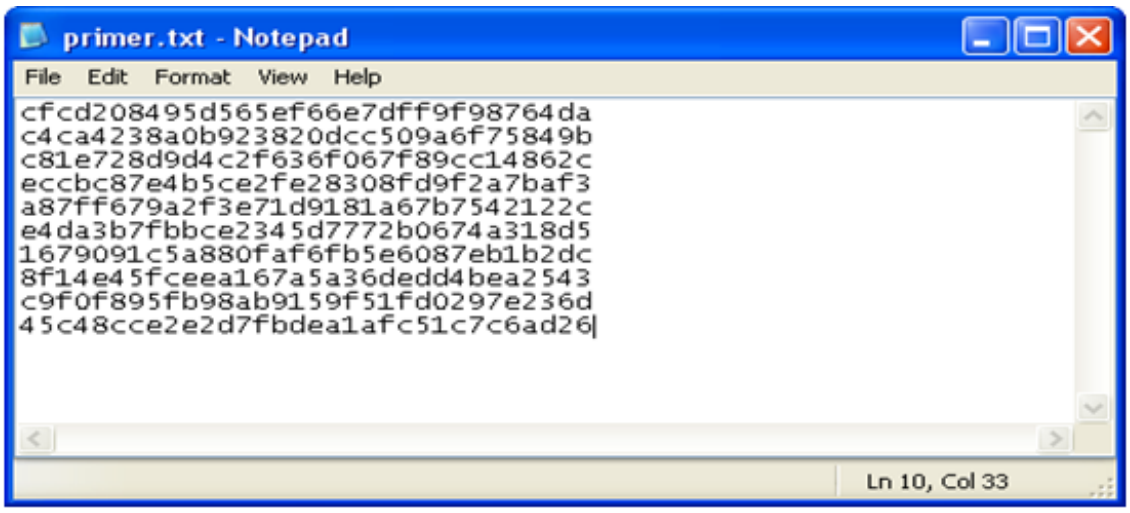

Рисунок 3.4 – Текстовый файл с хеш-шифрами

При этом выбираем в hash type – MD5, и в Line format – hash (рисунок 3.6). Импортируем файл ( ) primer.txt в PasswordsPro (рисунок 3.5).

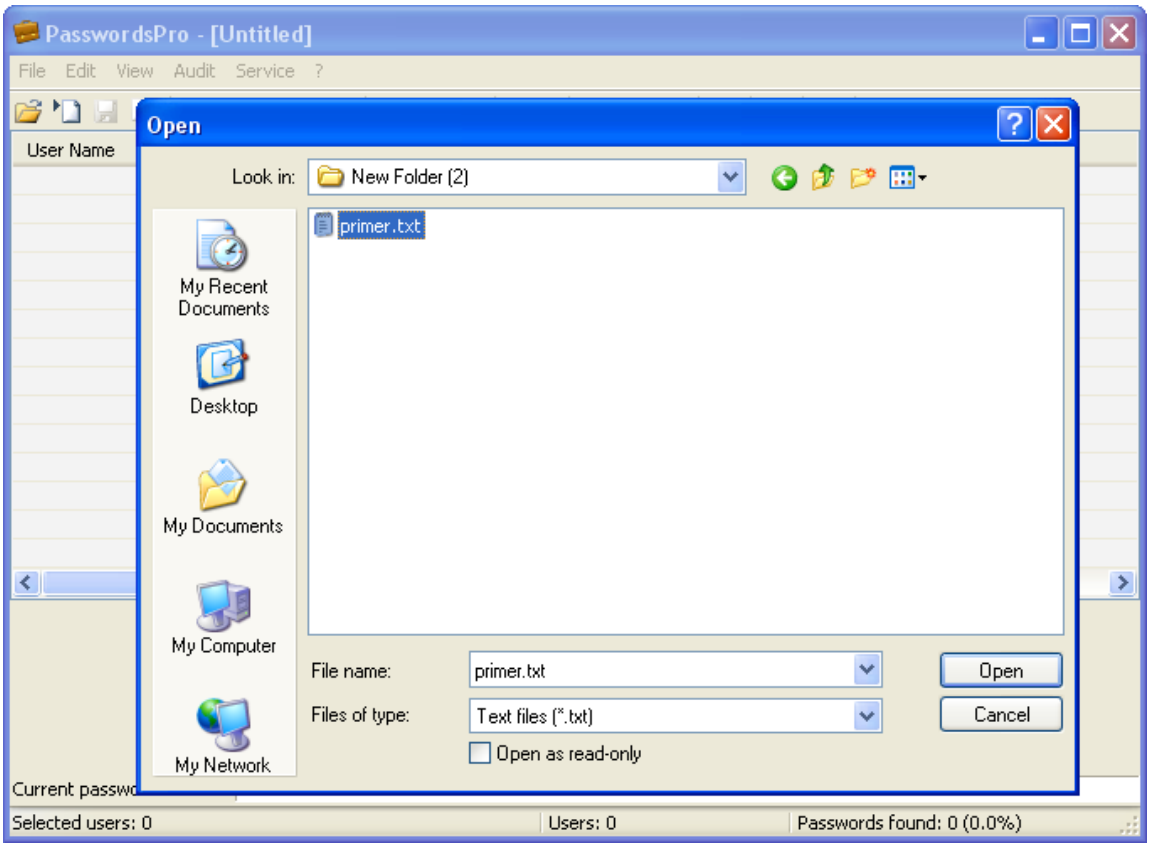

Рисунок 3.5 – Импорт таблицы хэша

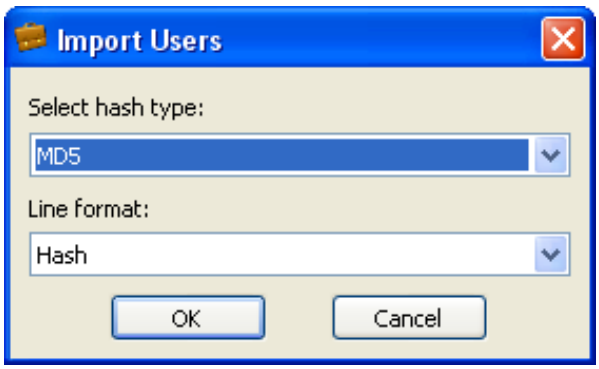

Рисунок 3.6 – Выбор параметров импортируемых данных

В настройках attack type ( ) выбираем Preliminary attack. Нажимаем Run attack from start ( ).

После окончания процедуры взлома появится окошко с паролями (рисунок 3.7).

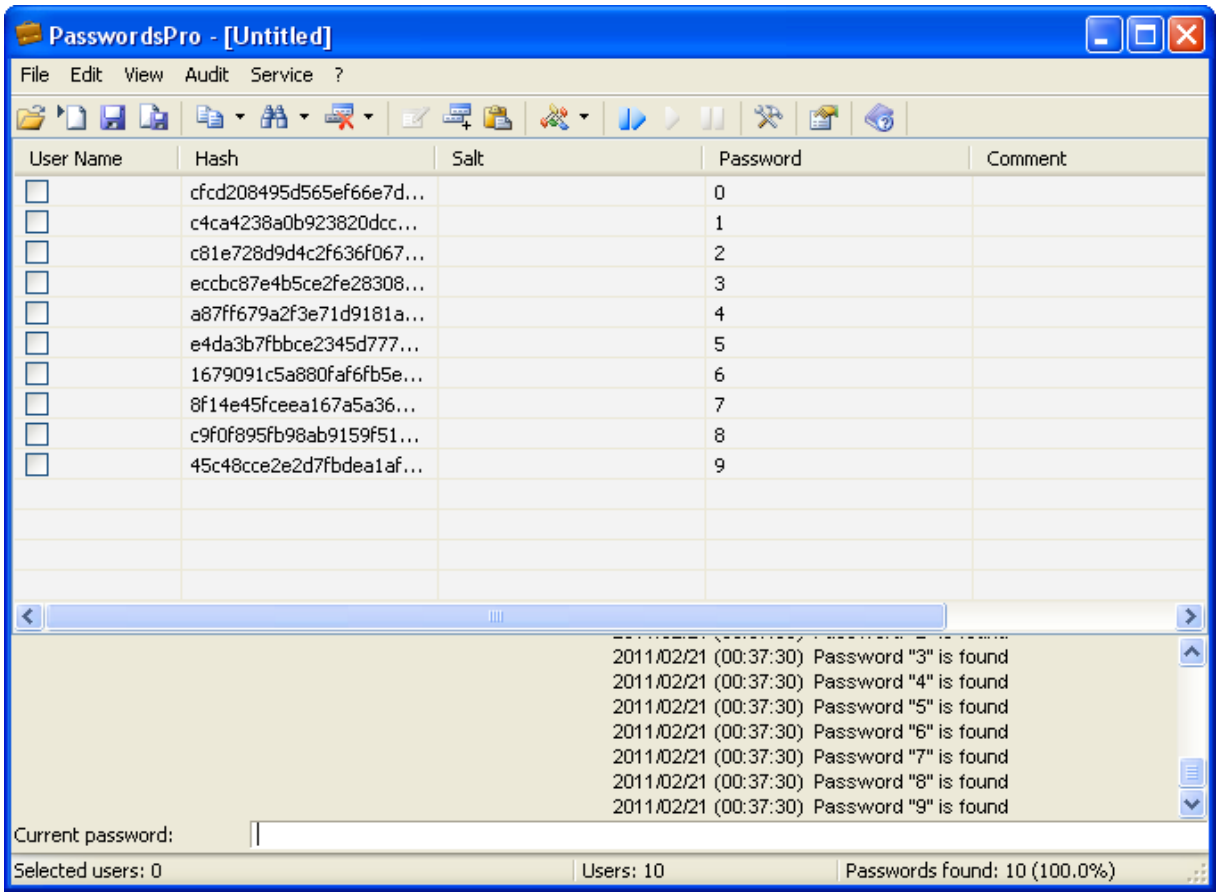

Рисунок 3.7 – Итоговое диалоговое окно с паролями

Алгоритм MD5, основанный на специфике использования методов криптоанализа, используемые сегодня, за счет использования алгоритмов и приложений для того, чтобы иметь доступ к обучению, это должно быть отмечено. Так, например, тема «Информационная безопасность» лабораторного

использования выводов этой работы. Асимметричная алгоритм шифрования является довольно распространенным типом образом. Радуга стол и криптоанализа с использованием других методов сегодня и расшифровать данные, которые они считают уязвимость современных информационных сетей и других методов, защита конфиденциальной информации и практических навыков может ввести различные студентов.

1) Изучение алгоритма MD5 и программного обеспечения генератора, является MD5 и PasswordsPro использовать один пример. Описание работы и их интерпретация представлены в виде шифрования и криптоанализа MD5 с простых задач.

Алгоритм MD5 современный криптография уверен укоренились и (равный-одноранговой сети, пароль пользователя на сервере, например, новостей и информации) информации ограниченного к аутентификации пользователя и проверки потока используется почти все сетевой информации с использованием. Асимметричная алгоритм шифрования является довольно распространенным типом образом.

3.2 Криптоанализ алгоритма RSA с использованием хеш-функций

CrypTool 2 программное обеспечение, используя RSA исследования алгоритма. Пользователи уже знакомы с программой, а также рабочее окно масштабирования позволяет интуитивно Windows Presentation Foundation (WPF) на основе вектора, используя графический интерфейс пользователя. На рисунке 1 показаны карты в главном окне программы.

Cryptool Beta 2 поддерживает различные классического Цезаря, такие как шифрование, DES, AES, RSA (несправедливое и симметричный) Vigenere современная форма знакомы. CrypTool Бета 2 Тесты, разнообразие дать Crypto аналитические инструменты, даже классические и современные шифры (например, кнопка самый простой метод для перебора) поврежден. Crypto анализ шифрования проводилось. Важность роли стандартов, таких как преобразования, такие как (текста), или булевой алгебры имеет и другие функции, такие как расширенная использования

Все шифры, используемые в программе были и аксессуары функциональная форма блок. Модуль CrypTool 2 это он задать соответствующие параметры для визуального программы предоставляет графический пользовательский интерфейс. Часть их может визуализировать внутренние операции. Криптографический алгоритм, все детали шифра блок сценария и смонтировать картину, так как он используется в реальной жизни, делает его удобным для пользователя.

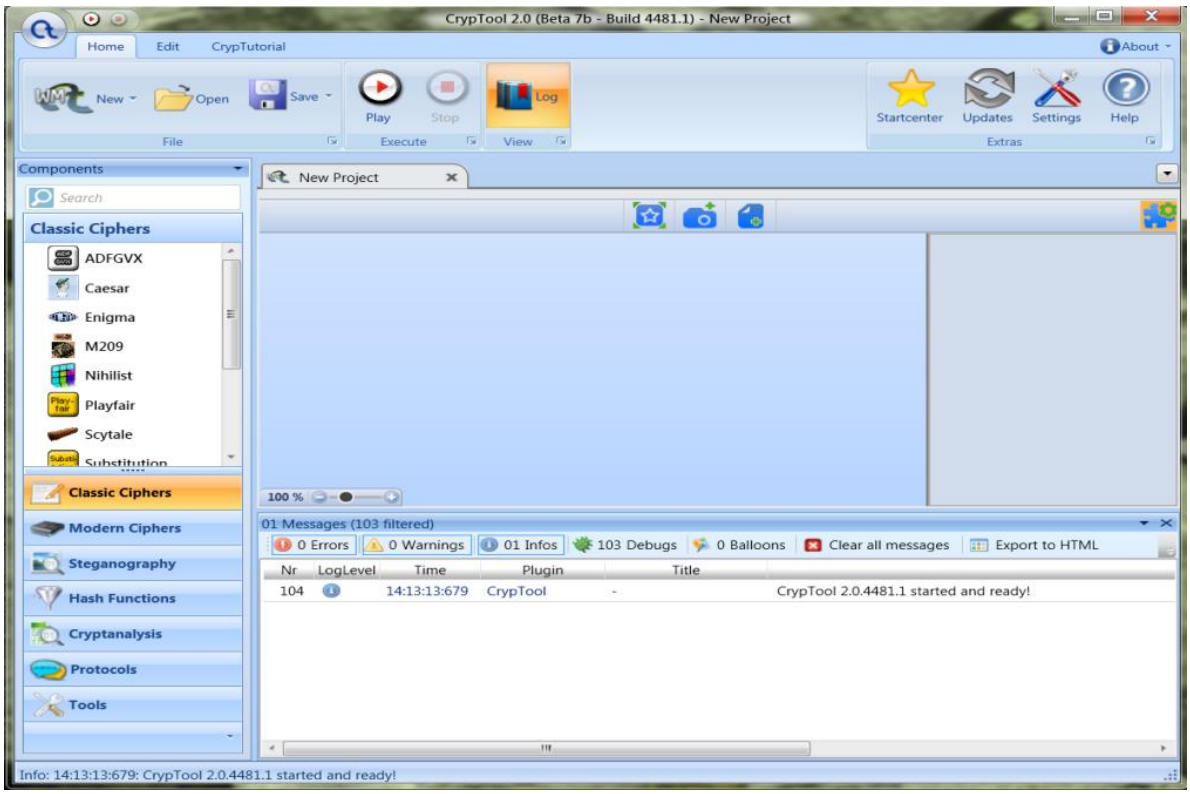

Рисунок 3.8 – Основное окно программной среды CrypTool 2 Beta

Функционал блоки имеют модули для ввода и вывода информации в процессе работы, которые в свою очередь могут присоединяться к другим функционал блокам и обмениваться информацию между собой.

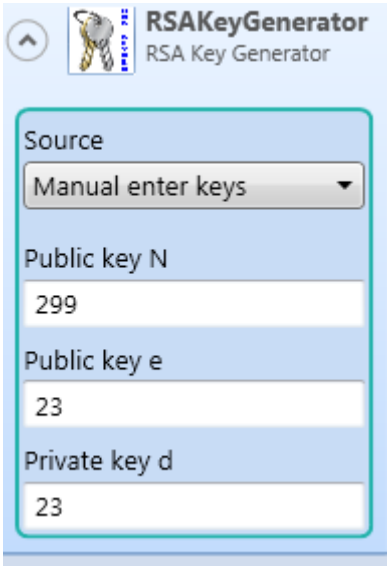

Рисунок 3.9 – Значения с параметрами ключа

Каждый блок имеет сценарий работы и виртуализации. Это дает возможность после сборки всей схемы запустить симуляцию работы. После запуска симуляции каждый из блоков начинает постепенную загрузку с

отображением процесса в виде процентного выполнения. По окончанию на каждом блоке и а также в отчетном окне отображается полный ход действий, в котором могут отображаться ошибки и не состыковки блоков в ввиду передаваемой информации. Это облегчает работу преподавателя и позволяет пользователю самостоятельно разобраться в ошибке.Так же пользователь прослеживает в процессе симуляции все данные на входе выходе каждого блока простым наведением курсора.

3.3 Реализация процесса шифрования алгоритма RSA с помощью CrypTool 2

Осуществим реализацию процесса шифрования, применив программный продукт CrypTool 2. Для этого соберем функциональную схему, представленный на рисунке 3.10.

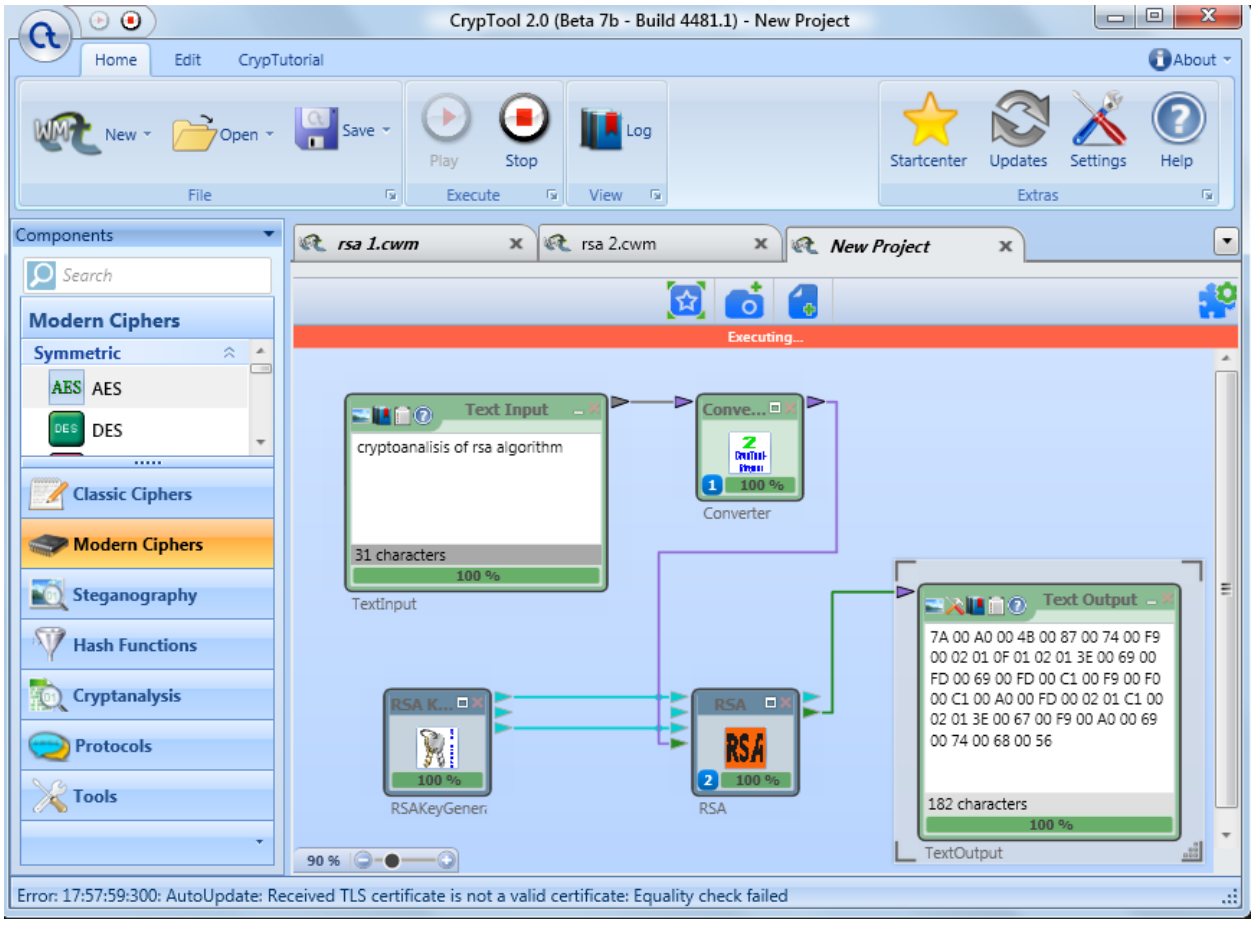

Рисунок 3.10– Функциональная схема процесса шифрования RSA

Для шифрования рассчитаем пару ключей по методике алгоритма RSA.

Одним из наиболее распространенных методов несимметричного шифрования- дешифрования является метод шифрования с открытым ключом, в котором используется алгоритм RSA.

Алгоритм основан на использовании операции возведения в степень модульной арифметики. Его можно представить в виде следующей последовательности шагов:

Шаг 1.

Выбирается два больших простых числа числа p и q. Простыми называются числа, которые делятся на самих себя и на 1. На практике для обеспечения криптостойкости системы величина этих чисел должна быть длиной не менее двухсот десятичных разрядов.

Шаг 2.

Вычисляется открытая компонента ключа n по формуле:

$$n = p\ q \qquad (3.1)$$

Шаг 3.

Находится функция Эйлера по формуле:

$$f(p, q)=(p-1)(q-1) \qquad (3.2)$$

Функция Эйлера показывает количество целых положительных чисел от1 до n, которые не имеют ни одного общего делителя, кроме 1.

Шаг 4.

Выбирается число e, которое должно взаимно простым со значением функции Эйлера и меньшим, чем f(p q.)

Шаг 5.

Определяется число d, удовлетворяющее соотношению:

$$e * d(\mathrm{mod}\ f(p\ q.))=1 \qquad (3.3)$$

Числа e и n принимаются в качестве открытого ключа. В качестве секретного ключа используются числа d и n. Примем несколько вариантов значения ключа и рассчитаем параметры закрытого и открытого ключа с помощью программы, представленной на рисунках 3.11-3.13:

1) p=251, q=31.
2) p=1523, q=113.
3) p=165173, q=1231

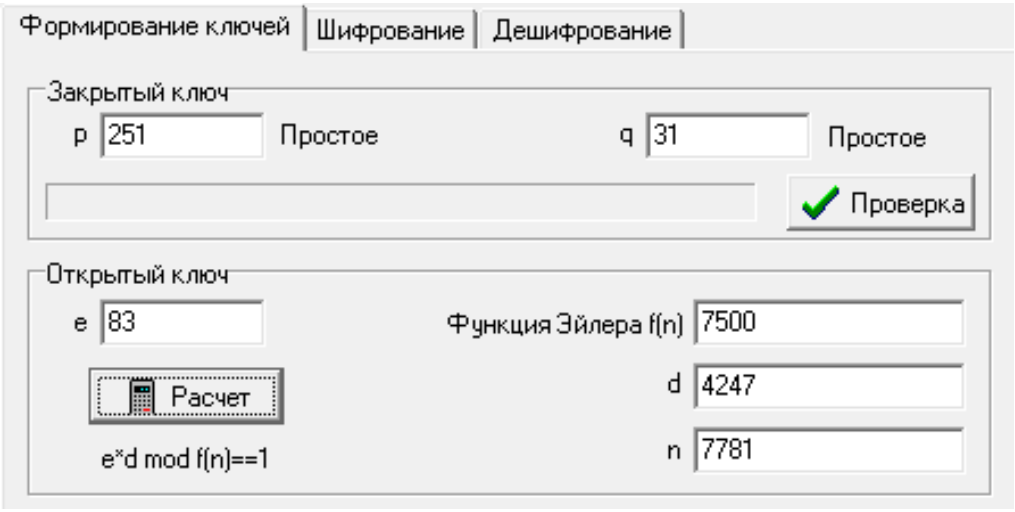

Рисунок 3.11 – Расчет ключа с параметрами 1

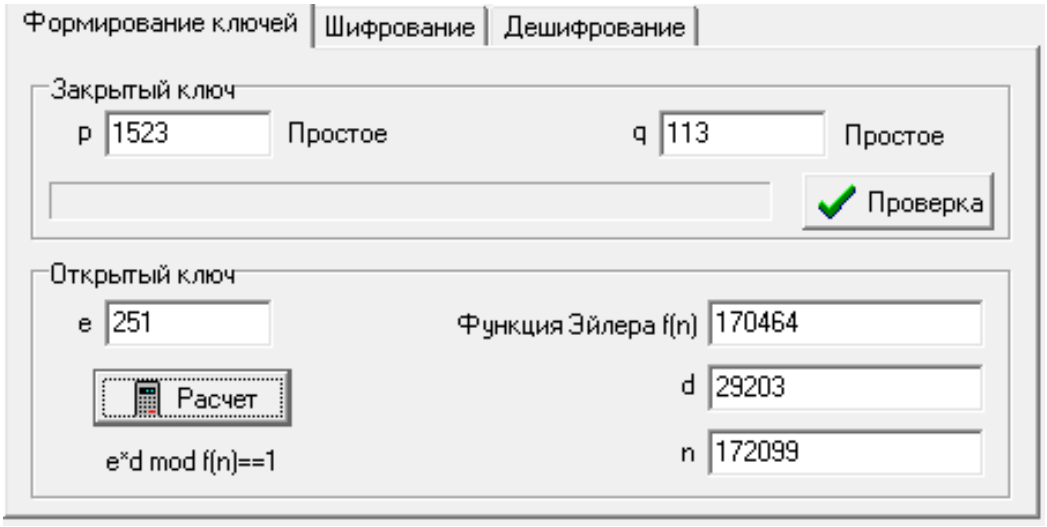

Рисунок 3.12 – Расчет ключа с параметрами 2

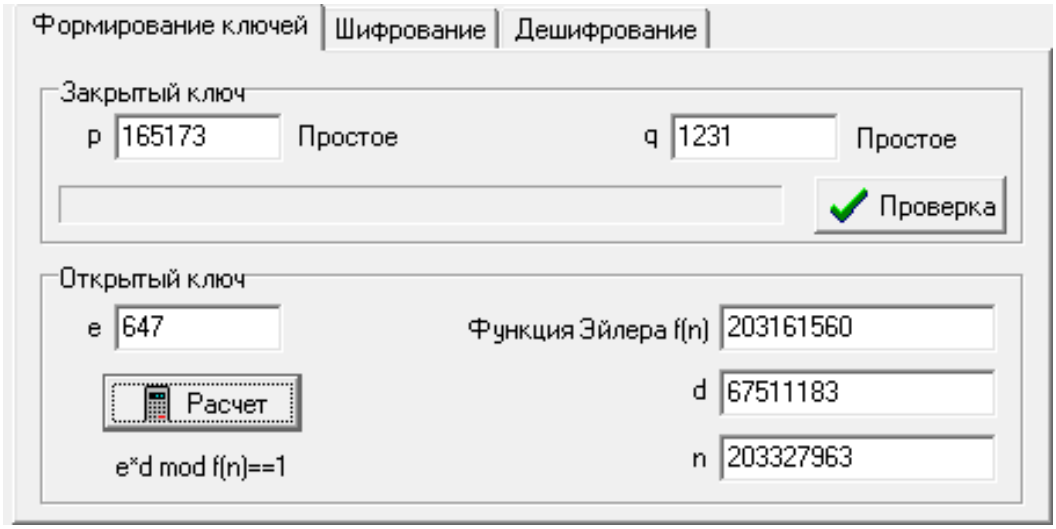

Рисунок 3.13 – Расчет ключа с параметрами 3

Таблица 3.1 – Ключи шифрования

Параметр p, q	Открытый ключ	Закрытый ключ
(251, 31)	(83, 7781)	(4247, 7781)
(1523,113)	(251, 172099)	(29203, 172099)
(165173, 1231)	(647, 203327963)	(67511183, 203327963)

Внесем рассчитанные параметры в программу и произведем шифрование.

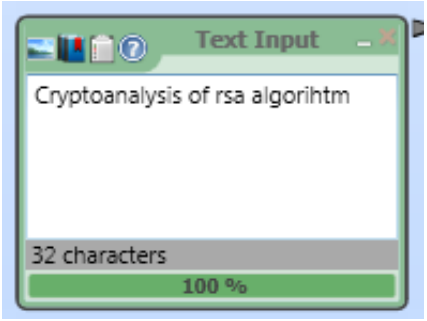

Рисунок 3.14 – Исходный текст

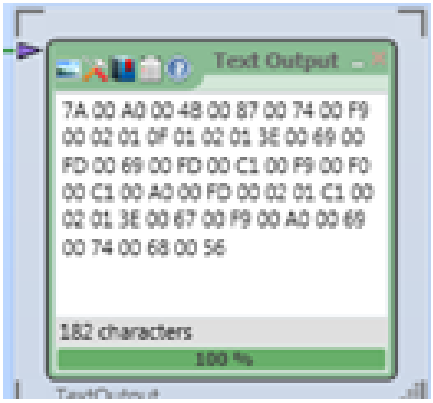

Рисунок 3.15 – Шифрованный текст

3.4 Шифрование с использованием хеш-функций

Построим функциональную схему шифрования с добавлением блока хеш-функции MD5 и произведем шифрование с теми же параметрами ключа (рисунок 3.16).

Построим функциональную схему шифрования с применением хеш-функции SHA-1c добавлением блока хеш-функции и произведем шифрование с теми же параметрами ключа (рисунок 3.17).

Шифрование с применением хеш-функции SHA-256 представлено на рисунке 3.18.

Шифрование с применением хеш-функции SHA-512 представлено на рисунке 3.19.

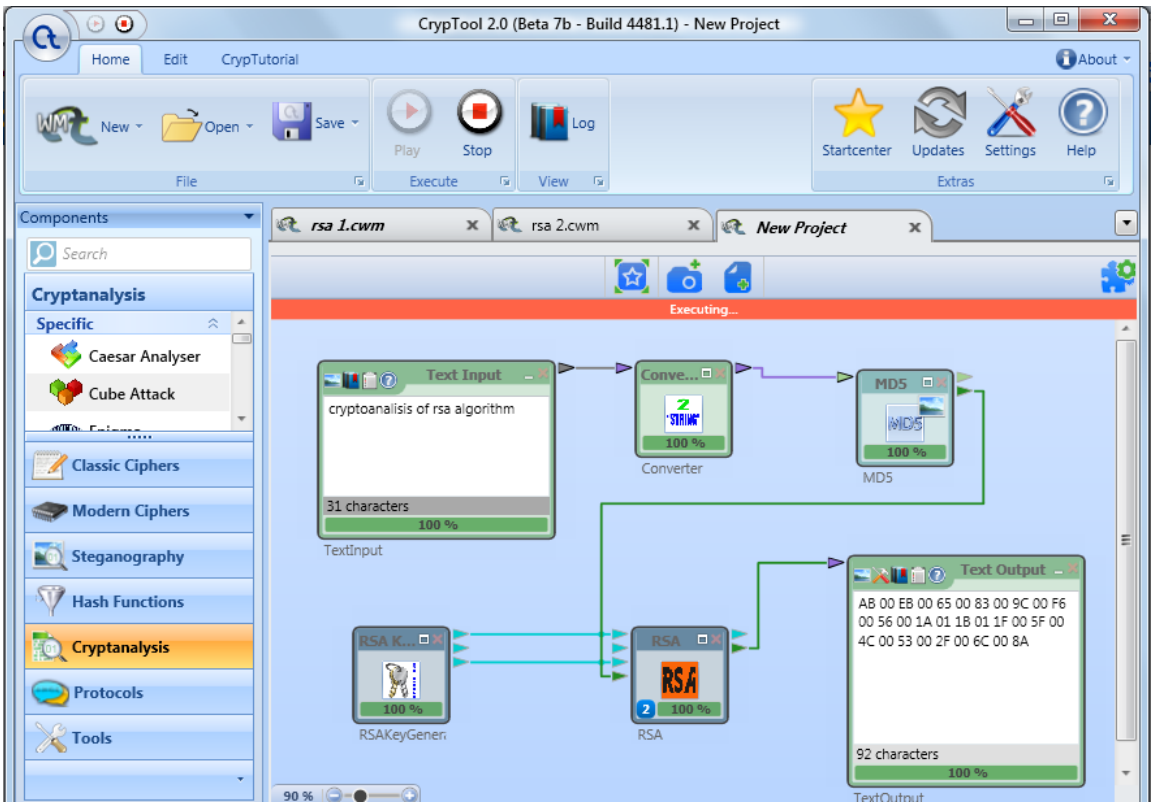

Рисунок 3.16 – Шифров ание с применением хеш-функции MD5

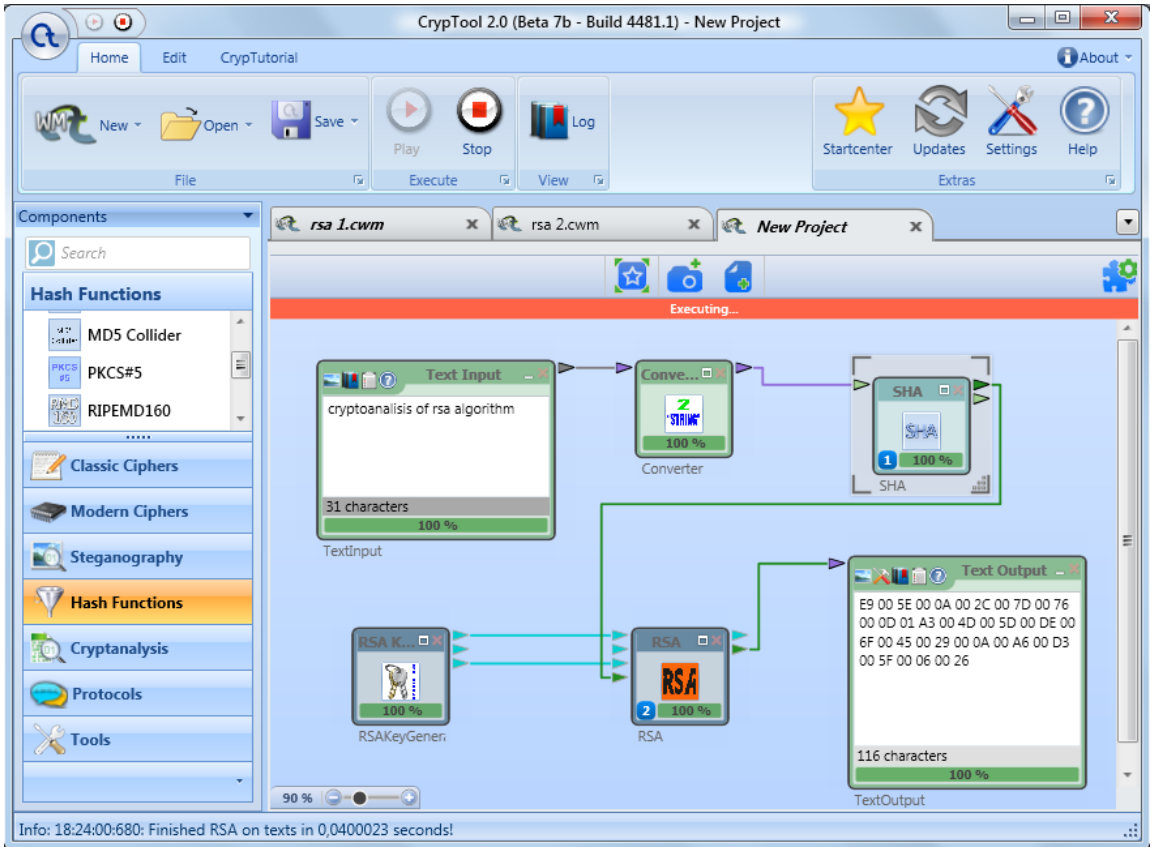

Рисунок 3.17 – Шифрование с применением хеш-функции SHA-1

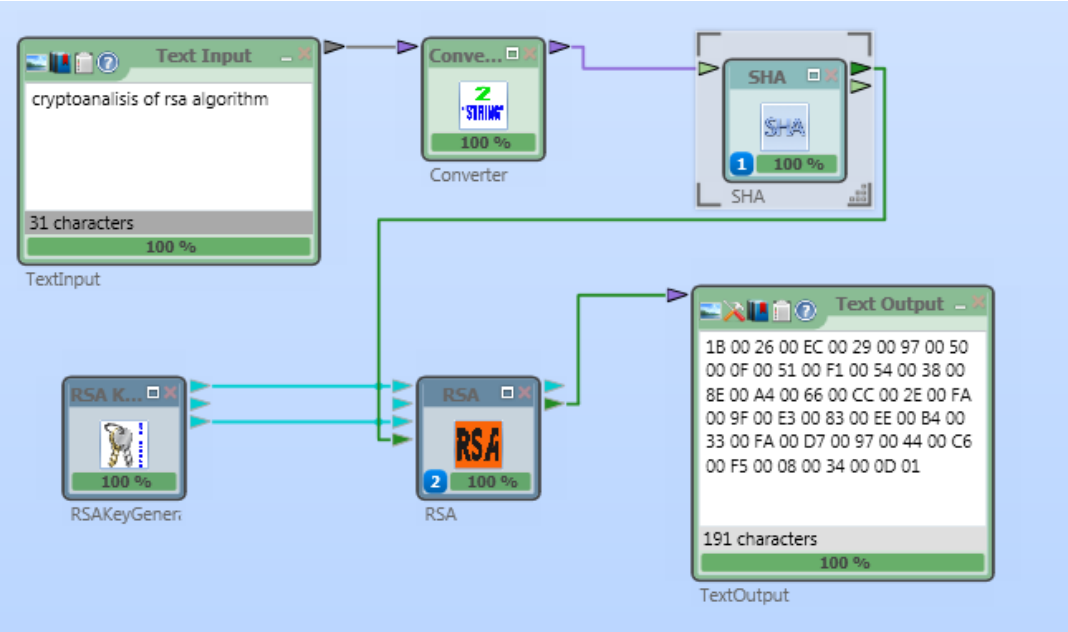

Рисунок 3.18 – Шифрование с применением хеш-функции SHA-256

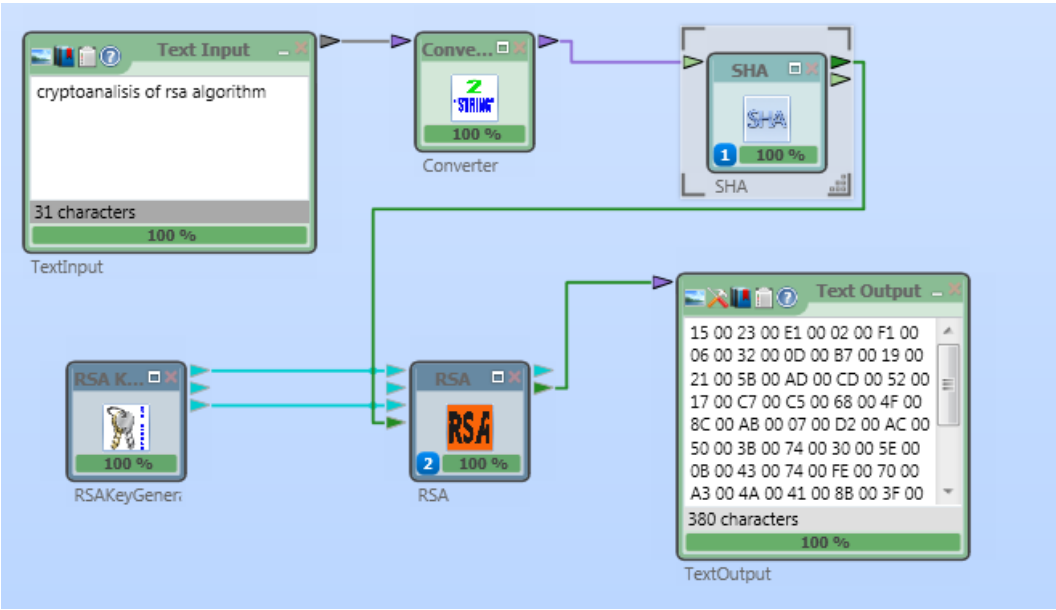

Рисунок 3.19 – Шифрование с применением хеш-функции SHA-512

Таблица 3.2 – Значения шифрованных символов

Алгоритм хеш-функции	Параметр ключа	Параметр шифрования, символы
RSA	(23,299) - открытый (23,299) - закрытый	182
MD5		92
SHA-1		116
SHA-256		191
SHA-512		380

Таблица 3.3 – Значения шифрованных символов

Алгоритм хеш-функции	Параметр ключа	Параметр шифрования, символы
RSA	(83,1643) - открытый (827,1643) - закрытый	185
MD5		92
SHA-1		119
SHA-256		191
SHA-512		380

Таблица 3.4 – Значения шифрованных символов

Алгоритм хеш-функции	Параметр ключа	Параметр шифрования, символы
RSA	(251,7781) - открытый (6859,7781) - закрытый	185
RSA + MD5		92
RSA + SHA-1		119
RSA + SHA-256		191
RSA + SHA-512		383

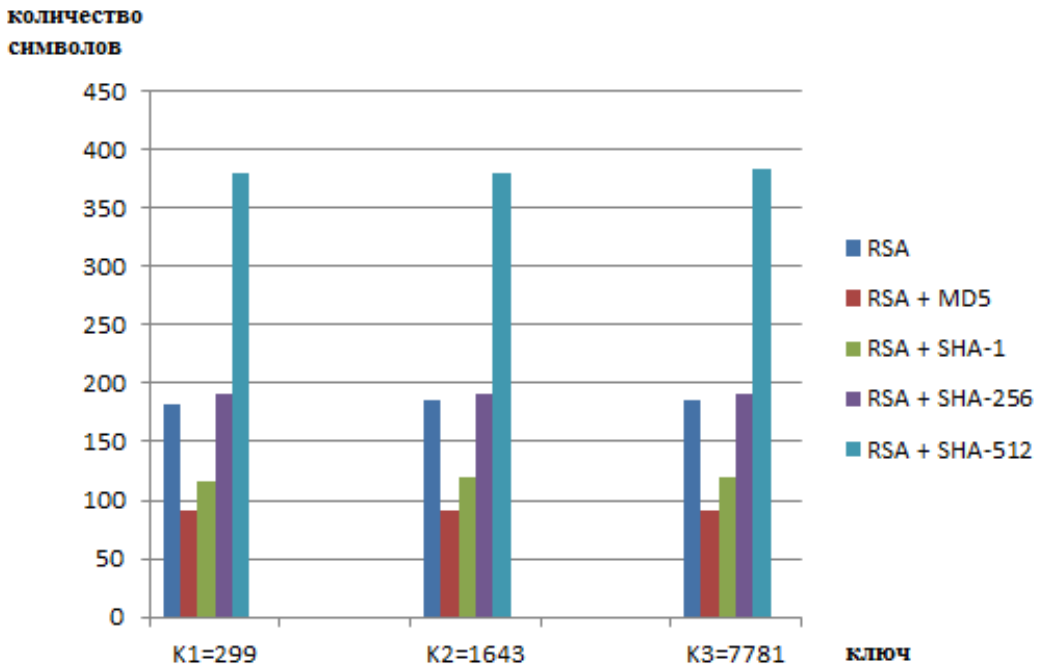

Рисунок 3.20 – Зависимость шифрованных символов от размера ключа

Из рисунка видно, что при увеличении размера ключа шифрованные символы алгоритма RSA с использованием SHA-512 увеличиваются. Так же рисунок показывает эффективность использования алгоритмов шифрования. Можно сделать вывод, что наиболее экономичный из алгоритмов – это с использованием хеш функции RSA + MD5. Если не использовать хеш-

функцию, что алгоритм шифрования имеет почти в 2 раза больше шифрованных символов в отличие от исходного текста.

Таблица 3.6 – Значения шифрованных байт

Алгоритм хеш-функции	Параметр ключа	Параметр шифрования, байт
MD5	(23,299) - открытый (23,299) - закрытый	0
SHA-1		233
SHA-256		27
SHA-512		21

Таблица 3.7 – Значения шифрованных байт

Алгоритм хеш-функции	Параметр ключа	Параметр шифрования, байт
MD5	(83,1643) - открытый (827,1643) - закрытый	33
SHA-1		38
SHA-256		130
SHA-512		148

Таблица 3.8 – Значения шифрованных байт

Алгоритм хеш-функции	Параметр ключа	Параметр шифрования, байт
RSA + MD5	(251,7781) - открытый (6859,7781) - закрытый	33
RSA + SHA-1		163
RSA + SHA-256		36
RSA + SHA-512		94

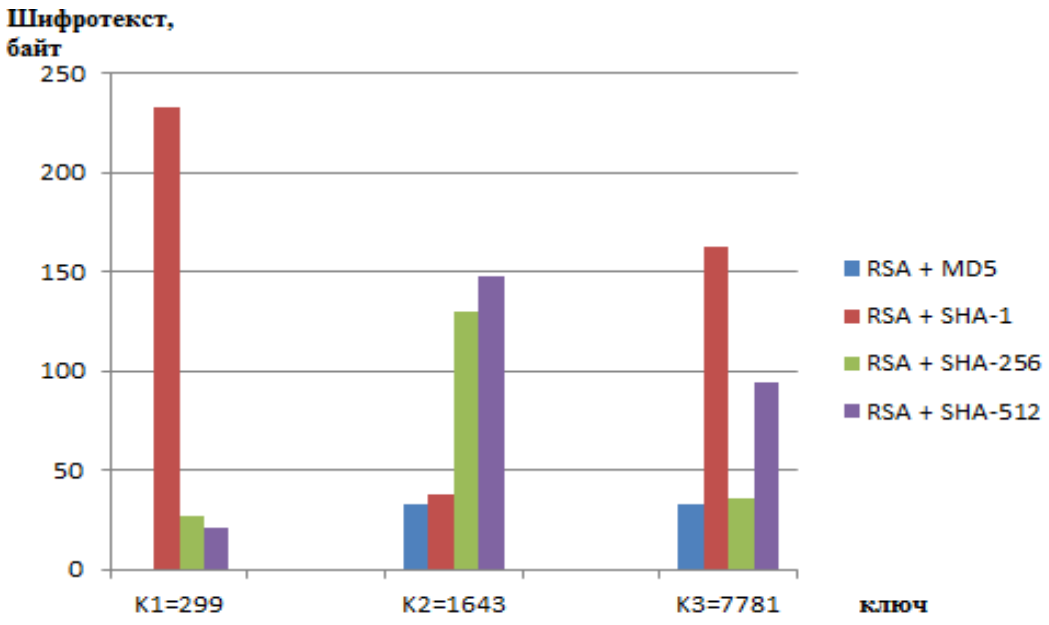

Рисунок 3.21 – Зависимость объема шифротекста от размера ключа

Рисунок 3.21 показывает количество шифрованных битов в зависимости от размерности ключа. При использовании ключа 1 шифрование по алгоритму RSA с использованием хеш-функции MD5 невозможна, так как исходный текст меньше размерности ключа. В таком случае шифрование с использованием данного метода не целесообразно.

3.5 Атаки на алгоритм RSA

Для дешифрации необходимо по известным N, e и шифртексту y найти такое $x \in ((Z/N))^*$, что $y = x^e \bmod N$.

Попытаемся решить сравнение при конкретных y, затем использовать гомоморфность отображения D(x).

Один из возможных способов следующий: пусть имеется набор пар $\{(x_1, y_1)...(x_k, y_k)\}$ с условием, что $x_i^e = y_i \bmod N$, 1 < y < N, (y, N) = 1. Если каким-либо образом удалось представить y в виде $y = y_1^{s_1} \cdot ... \cdot y_k^{s_k} \bmod N$ с целыми s_k, то $x = x_1^{s_1} \cdot ... \cdot x_k^{s_k}$ будет решением сравнения $y = x^e \bmod N$.

В наличии имеется открытый ключ N = 31459, e = 5 и набор пар соответствующих друг другу исходных и зашифрованных сообщений: (23, 18707), (755, 26871), (631, 6384). Требуется расшифровать шифртекст y = 11 638. Для этого представим y в виде $y = 18\,707^{-1} \cdot 26\,871^3 \cdot 6\,384^{-2} = 11\,638.$ Отсюда легко вычислить исходное сообщение: $x = 23^{-1} \cdot 755^3 \cdot 631^{-2} = 28\,260$.

Заметим, что этот подход не менее труден, чем поиск алгоритма решения сравнения $y = x^e \bmod N$.

3.5.1 Взлом RSA при неудачном выборе параметров криптосистемы (Метод Ферма)

Само по себе использование RSA не обеспечивает безопасности. Дело еще в деталях реализации. Приведем ряд примеров. Для простоты вычислений будем работать с небольшими числами. Цель – показать особенности, не зависящие от размера [8].

Пусть пользователь выбрал N = 2047, e = 179, d = 411. Так как 2047 = 2389, а $\varphi(23) = 22$, $\varphi(89) = 88$ имеют наименьшее общее кратное 88, то любой обратный к 179 по модулю 88, например 59, будет действовать как d.

Число N = 536813567 является произведением простого числа Мерсенна 8191 и простого числа Ферма 65537. Это очень плохой выбор.

Число 23360947609 является очень плохим выбором для N из-за того, что два его простых делителя слишком близки к друг другу. Пусть p > q, тогда имеем $N = (\frac{p+q}{2})^2 + (\frac{p-q}{2})^2$. Обозначим: $t = \frac{p+q}{2}$, $S = \frac{p-q}{2}$. Так как S мало, то t – целое число, лишь немного большее $\sqrt{N}$, причем t^2 – N является полным квадратом. Проверяем подряд целые числа $t > \sqrt{N}$. В нашем примере t_1 =

152843, t_2 = 152844, t_3 = 152845 и $t^3 - N = 804^2$, тогда $p = 152\,845 + 804$, p = 152845 – 804. Таким образом, мы с третьей попытки нашли p и q. Количество попыток, необходимых для факторизации N, можно при известных p и q вычислить по следующей формуле: $k = \sqrt{p \cdot q + (\frac{p-q}{2})^2} - \left[\sqrt{p \cdot q}\right]$, где [x] – операция округления x до ближайшего целого числа.

Релизация атаки на алгоритм шифрования RSA посредством метода Ферма с помощью программы BCalc. Снимок экрана с окном программы «BCalc» приведен на рисунке 3.22.

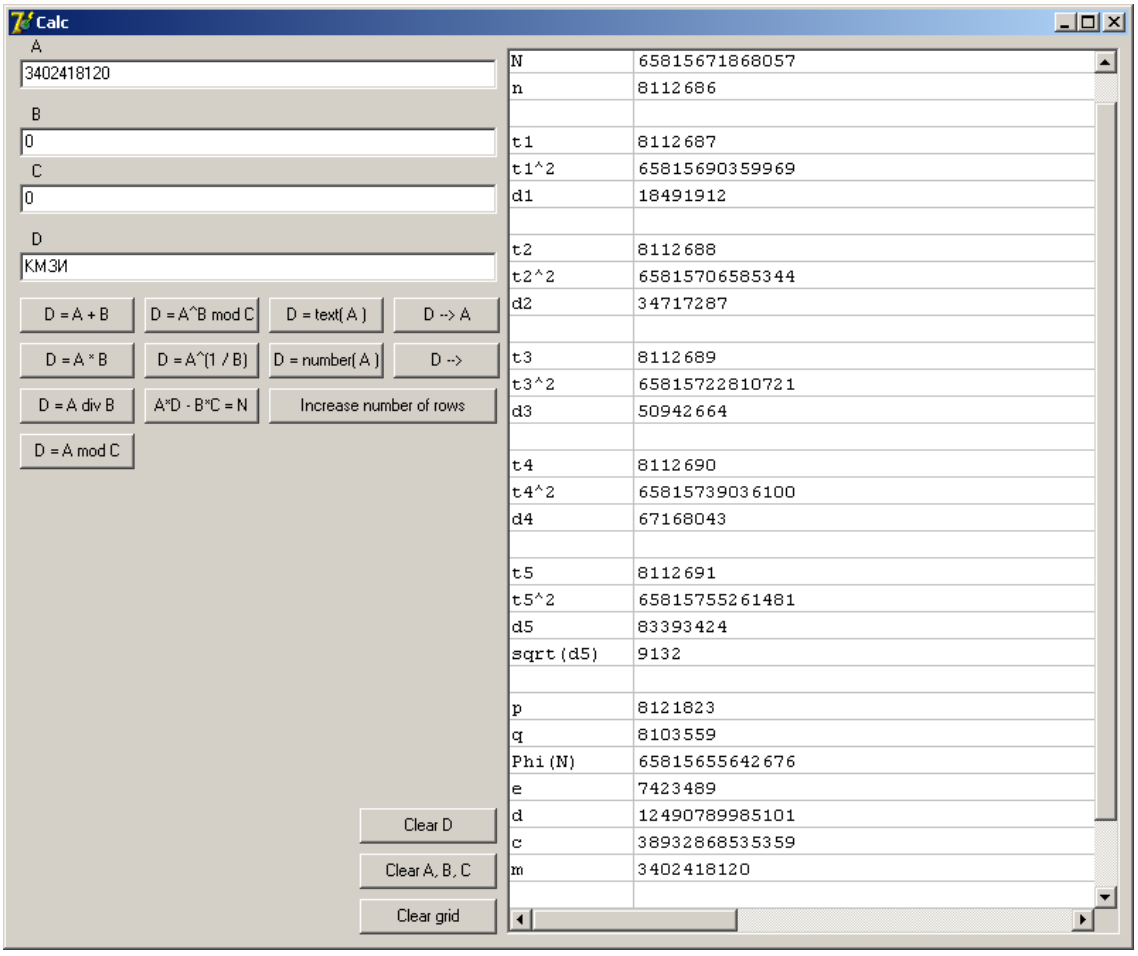

Рисунок 3.22 – Расчет алгоритма с помощью программы BCalc

Исходные данные:
N = 65815671868057;
e = 7423489;
C = 38932868535359.

Вычисляем n = [sqrt(N)] + 1. В поле A помещаем N, в поле B – 2; нажимаем кнопку «D = A^(1/B)». В поле D заносится число 8112686, в первую строку таблицы – сообщение «[error]». Это свидетельствует, о том, что N не является квадратом целого числа [8].

t1 = n + 1.

Возводим число t1 в квадрат:

A: = 8112687,

B: = 2,

C: = 0 (возведение в квадрат будет производиться не по правилам модульной арифметики), нажимаем «D = A^B mod C» => D = t1^2 = 65815690359969.

Вычисляем w1 = t1^2 – N. Для этого A:= t1^2, B:= –N, затем нажимаем «D = A + B» => D= w1 = 18491912. Проверяем, является ли w1 квадратом целого числа: A:= w1, B:= 2, нажимаем «D = A^(1/B)» => в первой строке таблицы появляется сообщение «[error]», следовательно проделываем п. 2 заново с t2 = n + 2 и так далее, пока не найдем, что некое wi является квадратом целого числа.

При вычислении квадратного корня w5 первая строка таблицы остается пустой, а D = sqrt(w5) = 9132, что свидетельствует об успехе факторизации. t5 = 8112691.

Вычисляем:

p = t5 + sqrt(w5);

A:= t5, B:= sqrt(w5), нажимаем «D = A + B» => D = p = 8121823; q = t5 – sqrt(w5) = 8103559.

Вычисляем:

Phi(N) = (p – 1)(q – 1),

A:= 8121822,

B:= 8103558, нажимаем «D = A·B» => D= Phi(N) = 65815655642676.

Вычисляем d, как обратный к e:

A:= e,

B:= –1,

C:= Phi(N), нажимаем «D = A^B mod C» => D = d = 12490789985101.

Производим дешифрацию шифрблока C: A:= C; B:= d; C:= N. Нажимаем «D = A^B mod C». В поле D находится исходное сообщение M = 3402418120. Переводим M в текстовый вид. Для этого A:= M, нажимаем «D = text(A)» => D = = «КМЗИ».

3.5.2 Атака повторным шифрованием

Строим последовательность: $y_1 = y,\ \ y_i = y_{i-1}^e (\mathrm{mod}\ N),\ i > 1$. Итак, $y_m = y^{e^m} (\mathrm{mod}\, N)$, а так как НОД$\left(e, \varphi(N)\right) = 1$, то существует такое натуральное число m, что $e^m \equiv 1 (\mathrm{mod}\, \varphi(N))$. Но тогда $y^{e^m - 1} \equiv 1 (\mathrm{mod}\, N)$, отсюда следует, что $y^{e^m} \equiv y (\mathrm{mod}\, N)$, значит, y_{m-1} – решение сравнения $y = x^e \left(\mathrm{mod}\, N\right)$.

Пусть у нас имеется открытый ключ N =84517 , e = 397 и зашифрованное им сообщение y = 8646. Необходимо найти исходный текст x. Возведем y в степень e и получим y_2 = 37043. Будем повторять операцию до тех пор, пока не получим y_n = y. y_{n-1} – искомое сообщение: y_3 = 5569, y_4 = 61833,

y_5 = 83891, y_6 = 16137, y_7 = 8646. y_6 является решением сравнения $y = x^e \pmod N$, а, следовательно, искомым сообщением x.

Анализ метода повторного шифрования хорошо показывает необходимость соблюдения требований на выбор p и q для обеспечения стойкости. В данном примере d = 82 225. Неудачный выбор криптосистемы привел к тому, что атака методом повторного шифрования дала результат почти сразу, тогда как нахождение d потребовало бы на порядок больших вычислений.

3.5.3 Атака на основе Китайской теоремы об остатках

Как отмечалось ранее, системы шифрования с открытыми ключами работают сравнительно медленно. Для повышения скорости шифрования RSA на практике используют малую экспоненту зашифрования.

Если вы выберите номер или E Малая, так что его двоичным представлением небольших подразделений может обрабатывать шифрование ¬ значительный импульс. Например, не выбрать e = 3 (не p - 1 и д - 1 не должна быть разделена на 3), шифрование может быть выполнена с помощью одного из ¬ воздвигнутый на площади модуля N и умножение. Выбор 65537 - количество двоичной записи, ¬ которая имеет только два устройства могут быть реализованы с использованием модуля шифрования N 16 squarings и одну не ¬ remnozheniya. Если экспонента e выбирается случайным, то реализация алгоритма шифрования RSA требует знания я повлиять на квадратную модуль с.ш., среднюю / 2 умножение в том же модуле, что на 5 - длина двоичного представления N. Однако, выбор меньшего показателя e может привести ¬ ти негативные последствия. Тот факт, что несколько корреспондентов же показатель является.

Пусть, например, три корреспондента имеют попарно взаимно простые модули N_1, N_2, N_3 и общую экспоненту e = 3. Если еще один пользователь посылает им некое циркулярное сообщение x, то криптоаналитик противника может получить в свое распоряжение три шифрованных текста $y_i = x^3 (\mathrm{mod}\ N_i)$,
i = 1, 2, 3. Далее он может найти решение системы сравнений, лежащее в интервале $0 < y < N_1 \cdot N_2 \cdot N_3$

$$\begin{cases} y \equiv y_1 (\mathrm{mod}\ N_1), \\ y \equiv y_2 (\mathrm{mod}\ N_2), \\ y \equiv y_3 (\mathrm{mod}\ N_3), \end{cases}$$

По китайской теореме об остатках такое решение единственно, а так как $x^3 < N_1, N_2, N_3$, то $y = x^3$. Значение x можно найти, вычислив кубический корень $x = \sqrt[3]{y}$.

Отметим, что выбор малой экспоненты расшифрования d также нежелателен в связи с возможностью определения d простым перебором. Известно также что если $d < \sqrt[4]{N}$, то экспоненту d легко найти, используя непрерывные дроби.

Три пользователя имеют модули $N_1 = 26549$, $N_2 = 45901$, $N_3 = 25351$. Все пользователи используют экспоненту $e = 3$. Всем пользователям было послано некое сообщение x, причем пользователи получили сообщения $y_1 = 5366$, $y_2 = 814$, $y_3 = 4454$. Найдем $M_0 = N_1 \cdot N_2 \cdot N_3 = 30893378827799$. Далее находим:

$m_1 = N_2 \cdot N_3 = 1163636251$

$m_2 = N_1 \cdot N_3 = 673043699$

$m_3 = N_1 \cdot N_2 = 1218625649$

$n_1 = m_1^{-1} \bmod N_1 = 13533$

$n_2 = m_2^{-1} \bmod N_2 = 27930$

$n_3 = m_3^{-1} \bmod N_3 = 22354$

$S = y_1 \cdot n_1 \cdot m_1 + y_2 \cdot n_2 \cdot m_2 + y_3 \cdot n_3 \cdot m_3 = 84501028038745578 + 15301661957638980 + + 121332116653000684 = 221134806649385242$

$S \bmod M_0 = 1000000000$

$x = (S \bmod M_0)^{1/3} = 1000$ – исходное сообщение, отправленное пользователям.

Исходные данные:
$N_1 = 363542076673$;
$N_2 = 728740902979$;
$N_3 = 522993716719$;
$C_1 = 246562834516$;
$C_2 = 291375746601$;
$C_3 = 222724269731$.

Последовательно вычисляем следующие значения:

$M_0 = N_1 \cdot N_2 \cdot N_3 = 138555669564008119302694433926047373$;

$m_1 = N_2 \cdot N_3 = 381126913374147389205901$;

$m_2 = N_1 \cdot N_3 = 190130221862955939995887$;

$m_3 = N_1 \cdot N_2 = 264927981225542872108867$;

$n_1 = m_1^{\wedge} (-1) \bmod N_1 = 287993142707$;

$n_2 = m_2^{\wedge} (-1) \bmod N_2 = 106614970676$;

$n_3 = m_3^{\wedge} (-1) \bmod N_3 = 32171022265$;

$S = c_1 \cdot n_1 \cdot m_1 + c_2 \cdot n_2 \cdot m_2 + c_3 \cdot n_3 \cdot m_3 =$
$= 348678927964033379521816073840676890870123543 29$;

$S \bmod M_0 = 676756407950945035621737840 00$;

$M = (S \bmod M_0)^{\wedge}(1/e) = 4075154940$;

text(M) = «тень».

Ниже приведен снимок экрана с окном программы «BCalc».

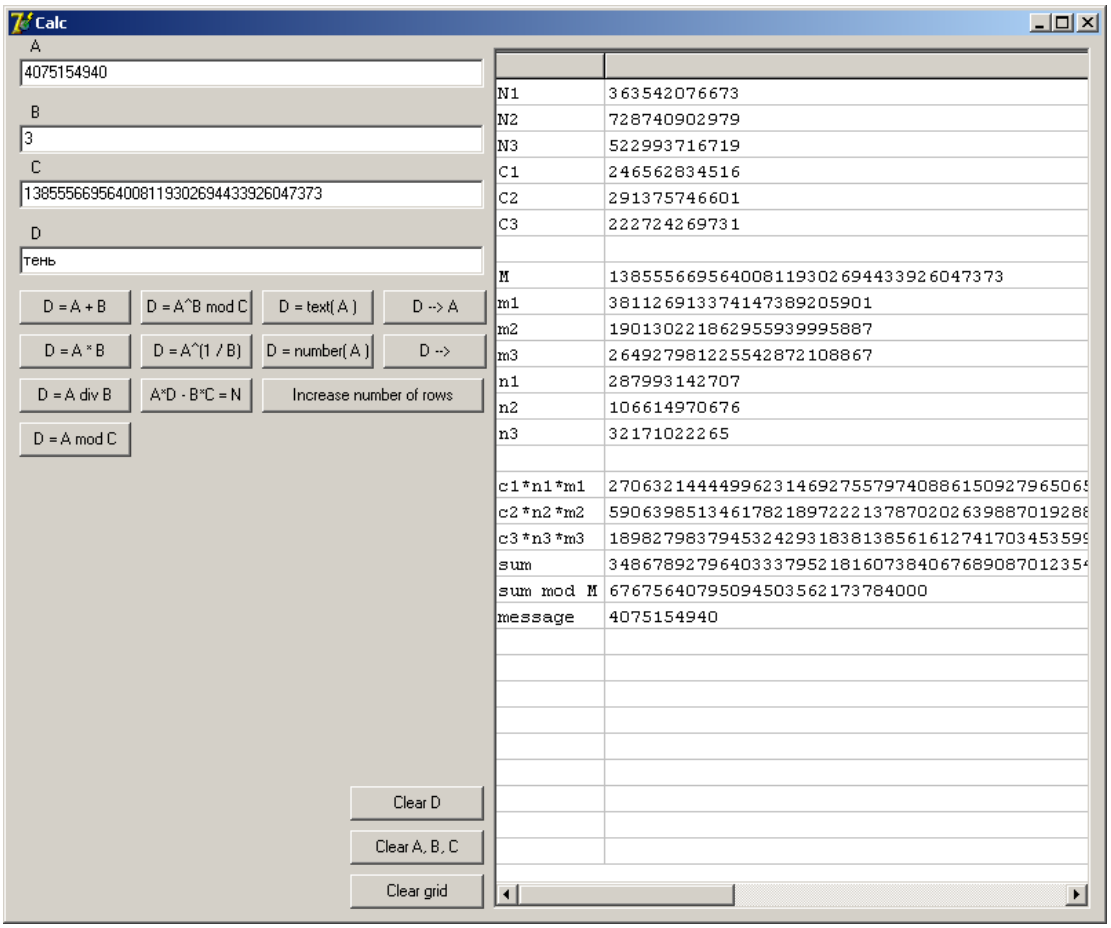

Рисунок 3.23 – Расчет алгоритма с помощью программы BCalc

3.5.4 Бесключевое чтение

Пусть два пользователя выбрали одинаковый модуль N и разные экспоненты e_1 и e_2. Если один пользователь посылает им некое циркулярное сообщение x, то криптоаналитик противника может получить в свое распоряжение два шифрованных текста $y_1 = x^{e_1} (\mathrm{mod}\, N)$ и $y_2 = x^{e_2} (\mathrm{mod}\, N)$. В таком случае криптоаналитик может получить исходное сообщение, используя расширенный алгоритм Евклида, находим r, s такие, что $re_1 + se_2 = 1$. Отсюда получаем: $y_1^r y_2^s = x^{re_1 + se_2} = x$

Два пользователя применяют общий модуль N = 137759, но разные взаимно простые экспоненты e_1 = 191 и e_2 = 233. Пользователи получили шифртексты y_1 = 60197 и y_2 = 63656, которые содержат одно и то же сообщение. Найдем исходное сообщение методом бесключевого чтения. Так как e_1 и e_2 взаимно просты, то найдем такие r и s, что $re_1 + se_2 = 1$. С помощью расширенного алгоритма Евклида находим r = 61, s = –50. Искомое сообщение $x = y_1^r \cdot y_2^s = 60197^{61} \cdot 63656^{-50} = 1234$

Исходные данные:

N = 357114156277;
e_1 = 1025537;
e_2 = 722983;
C_1 = 68639736967;
C_2 = 204258645263.

1. Решаем уравнение $e_1 \cdot r - e_2 \cdot s = \pm 1$. Для этого в поле A помещаем значение e_1, в поле B – значение e_2. Нажимаем кнопку «A·D – B·C = N», затем – кнопку C = s = 406030; D = r = 286243.

2. Производим дешифрацию: c_1 возводим в степень r, а c_2 – в степень –s по модулю N, тогда c_1^r = 189703239311, c_2^(–s) = 104340380259.

После этого результаты перемножаем и получаем, что m^($e_1 \cdot r - e_2 \cdot s$) = = 19793708126073817161549. Далее берем модуль от полученного значения: (m^($e_1 \cdot r - e_2 \cdot s$) mod N) = 1381187873 и преобразуем в текст «RSA!».

Ниже приведен снимок экрана с окном программы «BCalc».

Как видно из приведенных выше примеров, выбор параметров криптосистемы является сложной задачей. Параметры, которые можно выбрать в соответствии с жесткими требованиями. В настоящее время существующие методы (и исходя из имеющейся в настоящее время вычислительные мощности) нападение на алгоритма и / или криптографической возможно только при плохой выбор параметров. Среди доказательств необходимо провести работы, чтобы обеспечить каждому пользователю уникальное значение, P, Q и уникальную ценность e, которые отвечают условиям, описанным выше.

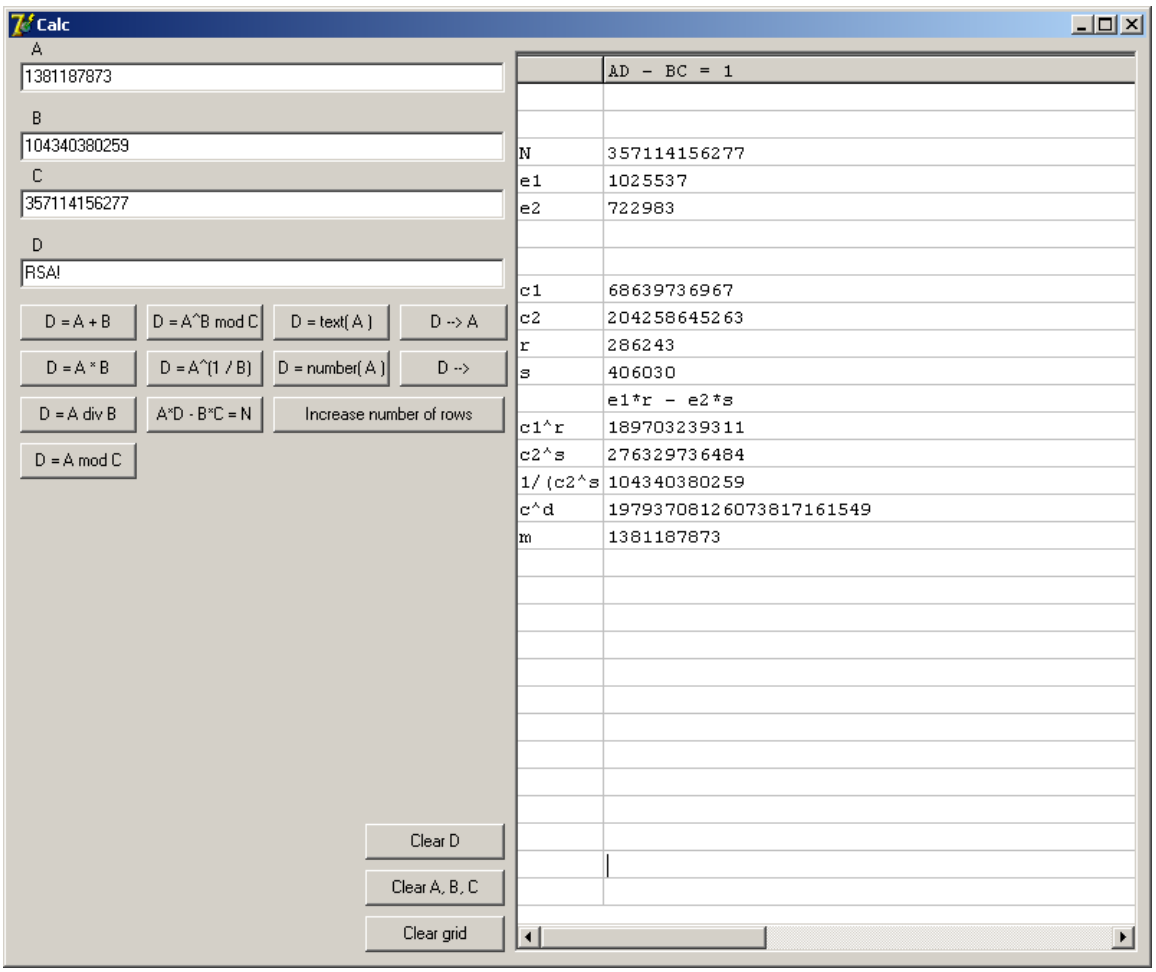

Рисунок 3.24 – Расчет алгоритма с помощью программы BCalc

Заключение

В данной магистерской диссертации были рассмотрены вопросы повышения криптостойкости алгоритмов шифрования. Для осуществления данного исследования используется программный продукт CrypTool 2.

Для исследования был выбран алгоритм RSA и произведено моделирование его алгоритма с помощью программного продукта Cryptool 2. Произведен расчет параметров ключа для алгоритма RSA с тремя значениями. Произведено моделирования алгоритма с учетом хеш-функций MD5, SHA-1, SHA-256, SHA-512 , а так же осуществлена атака по различным методам.

Так же в исследовании был представлен пример использования алгоритма MD5 и прикладного программного обеспечении Генератор MD5 и PasswordsPro. Работа представлена в виде простейших заданий шифрования и криптоанализа MD5, сопровождающихся иллюстрациями и пояснениями к ним.

В ходе исследований было выявлено, что при увеличении размера ключа шифрованные символы алгоритма RSA с использованием SHA-512 увеличиваются. Так же доказана эффективность использования алгоритмов шифрования.

Можно сделать вывод, что наиболее экономичный из алгоритмов – это с использованием хеш функции RSA + MD5. Если не использовать хеш-функцию, что алгоритм шифрования имеет почти в 2 раза больше шифрованных символов в отличие от исходного текста.

Список литературы

1. Баричев С. Г., Гончаров В. В., Серов Р. Е. Основы современной криптографии. — М.: Горячая линия — Телеком, 2002.
2. http://www.rsa.com/rsalabs/node.asp?id=2253 – Статья «What are MD2, MD4, and MD5?» - RSA Laboratories, 2000.
3. Гаммирование. Система RSA. Шифрование/расшифровывание. Сайт: http://www.studfiles.ru/dir/cat32/subj1166/file9284/view96935/page3.html
4. http://5ballov.qip.ru/referats/archive/27/5ballov-27299.zip
5. Пилиди В. С. Криптография. Вводные главы. — Ростов-на-Дону: ЮФУ, 2009.
6. http://study.online.ks.ua/spargalka/spori/inform-3.zip
7. Алгоритм DSA. Генерация ЭЦП.
http://www.studfiles.ru/dir/cat32/subj1166/file9284/view96935/page4.html
8. Алгоритм RSA: Методические указания к выполнению лабораторных работ http://window.edu.ru/resource/762/66762
9. http://5ballov.qip.ru/referats/archive/27/5ballov-27299.zip
10. Криптографическая хеш-функция MD5. http://dic.academic.ru/dic.nsf/ruwiki/18645
11. Скворцов А.В. Особенности реализации алгоритмов построения триангуляции Делоне с ограничениями // Вестник ТГУ. 2002. № 275. С.90–94.
12. http://www.studfiles.ru/dir/cat32/subj1166/file9284/view96935/page4.htm
13. Кнут Д. Искусство программирования, том 3. Сортировка и поиск. М.: Вильямс, 2000. 832 с.
14. Щербаков А., Домашев А. Прикладная криптография. М.: Русская редакция, 2003. 416 с.
15. Schieber R. Site Security Using Microsoft's CryptoAPI. N.Y.: Wrox, 2001. 49 p.
16. Bondi R. Cryptography for Visual Basic: A Programmer's Guide to the Microsoft CryptoAPI. N.Y.: John Weiley & Sons, 2000. 480 p.
17. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. М.: Триумф, 2002. 816 с.
18. Бронштейн И.Н., Семендяев К.А. Справочник по математике для инженеров и учащихся втузов. М.: Наука, 1986. 544 с.
19. IP Authentication using keyed MD5 by Bart Preneel (ESAT, K.U.Leuven, Belgium) Bart.Preneel@esat.kuleuven.ac.be
20. MD5 vs SHA-1, Performance and Pedigree by Masatake Ohta mohta@necom830.hpcl.titech.ac.jp
21. MD5 vs SHA by Ryan Malayter rmalayter@bai.org
22. www.secure-hash-algorithm-md5-sha-1.co.uk
23. www.webopedia.com
24. http://www.cryptool.org/- официальный сайт Cryptool.

25. Брассар Дж. Современная криптология. Мир ПК. №3. 1997.
26. http://software.intel.com/ru-ru/articles/advanced-encryption-standard-aes-instructions-set/
27. http://ru.wikipedia.org/wiki/Advanced_Encryption_Standard
28. Сыдыков Н.С. Исследование методов повышения криптографической стойкости. Поиск, №2(3), Алматы. – с.292-296.