Лабораторная работа № 3. Создание электронной подписи в документе

Цель работы:

Разработка процедур выработки и проверки электронной цифровой подписи (ЭЦП) сообщений на базе асимметричного криптографического алгоритма с применением функции хеширования

Постановка задачи:

1. Выбрать в соответствии с вариантом алгоритм вычисления хэш функции (контрольной суммы).
2. Реализовать программную реализацию алгоритма создания и проверки электронно-цифровой подписи.
3. Подписать текстовое сообщение.
4. Проверить правильность ЭЦП.
5. Внести изменения в сделанную подпись. Убедится, что подпись не является подлинной.
6. Результаты работы оформить в виде отчета

Описание используемого метода

Процесс генерации электронно-цифровой подписи:

1. Вычисляется хэш-код сообщения m: h=H(m), если $h \bmod q = 0$, то h присваивается значение 1.
2. Из диапазона [1,q] случайным образом выбирается значение k.
3. Вычисляется $r = a^{k} \bmod p$, $r1 = r \bmod q$; если r1=0, следует вернуться к предыдущему этапу и выработать другое значение k.
4. Вычисляется $s = (x * r1 + k * h) \bmod q$; если s=0, то необходимо вернуться к п.2 и выработать другое значение k.
5. Значения r1, s являются электронно-цифровой подписью сообщения m и передаются вместе с ним по каналам связи.

Проверка электронно-цифровой подписи:

1. Проверяется выполнение условий $0 < r1 < q$, $0 < s < q$, и если хотя бы одно из них нарушено, подпись отвергается.
2. Вычисляется хэш-код полученного сообщения m1: h=H(m1); если $h \bmod q = 0$, то h присваивается значение 1.
3. Вычисляется значение $v = h^{q-2} \bmod q$.
4. Вычисляются значения $z1 = (s * v) \bmod q$, $z2 = (q - r1) * v \bmod q$.
5. Вычисляется значение $u = (a^{z1} * y^{z2} \bmod p) \bmod q$.
6. Проверяем равенство u = r1. Если равенство выполняется, то подпись принимается. В противном подпись считается недействительной.

Описание исходных данных:

Параметры системы ЭЦП - числа p, q, a. Их можно выбрать самому с учетом ограничений на них: p = 2027, q = 1013, a = 2025 (p и q - простые числа, q является простым делителем числа p - 1, $1 < a < p - 1$, $a^{q} \bmod p = 1$).

Число x - секретный ключ для формирования подписи (1<x<q); его можно выбрать самому: x = 983

В данной работе я выбираю 1 вариант, поэтому у меня будет следующий алгоритм вычисления хэш функции (контрольной суммы): количество 1 в битовом представлении символов исходного текста.
Сообщение для которого мы генерируем ЭЦП:
"The divide-and-conquer algorithmic paradigm involves subdividing a large problem instance into smaller instances of the same problem."

Алгоритм работы программы:
Следует заметить, что в качестве вспомогательных алгоритмов, я использую алгоритм быстрого возведения в степень по модулю.
Алгоритм быстрого возведения в степень по модулю позволяет возводить число в некоторую степень (по некоторому модулю) со сложностью O(logN), используя двоичное представление степени.

В выбранном мной варианте используется следующий алгоритм вычисления хэш функции (контрольной суммы): количество 1 в битовом представлении символов исходного текста. Для этого мы преобразуем строку в массив байт в кодировке UTF-8, для каждого байта считаем количество 1 в его битовом представлении и суммируем результат.

Генерация ЭЦП:
1. Преобразуем исходное сообщение (заданное в виде строки) в массив байт в кодировке UTF-8. Вычисляем для полученного массива байт значение хэш функции (контрольной суммы).
2. Если вычисленное значение хэш функции (контрольной суммы) по модулю q равно 0, то присваиваем ему значение 1.
3. Инициализируем генератор псевдослучайных чисел.
4. Из диапазона [1,q] случайным образом выбираем значение k.
5. Вычисляем значения $r = a^{k} mod\, p$ и $r1 = r\, mod\, q$. Если r1 = 0, то возвращаемся в п.4.
6. Вычисляем $s = (x * r1 + k * h)\, mod\, q$; если s=0, то возвращаемся в п.4.
7. Мы сформировали ЭЦП: (r1, s). Возвращаем ее.

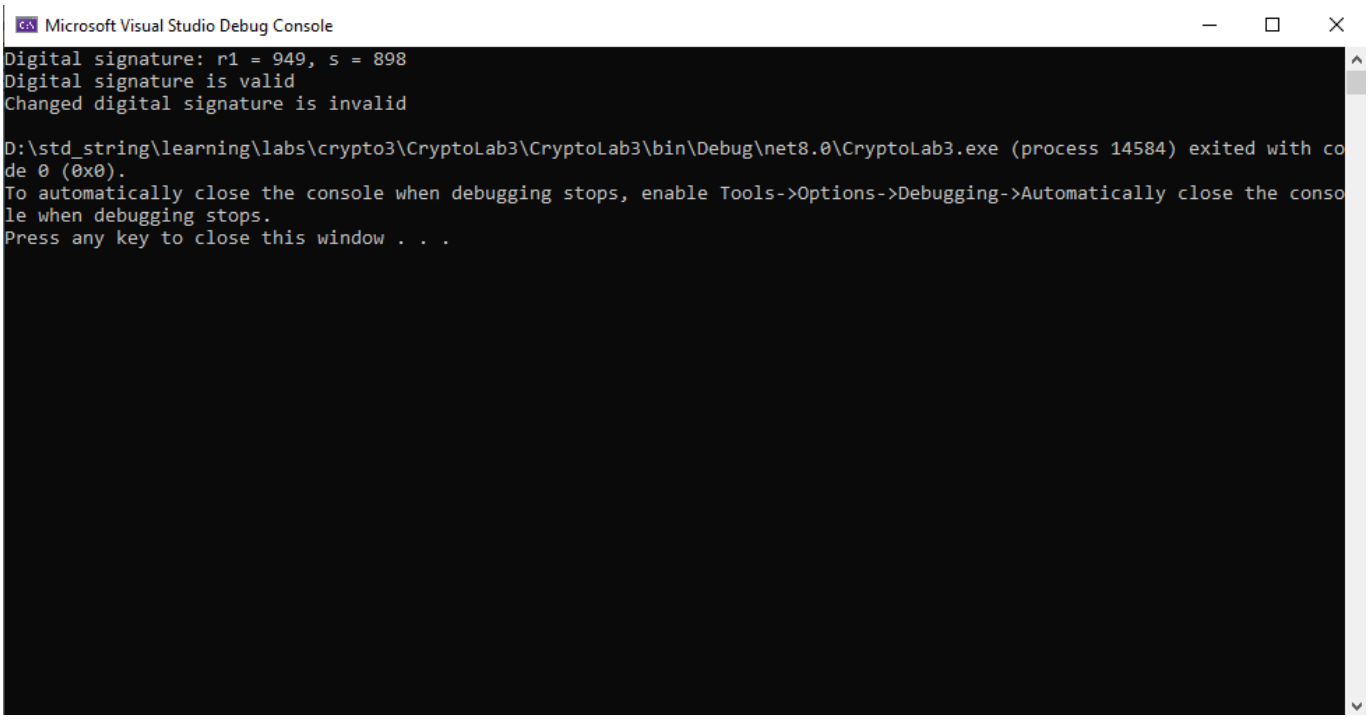
Проверка ЭЦП:
1. Проверяем выполнение условий 0 < r1 < q, 0 < s < q. и если хотя бы одно из них нарушено, то отвергаем подпись.
2. Преобразуем исходное сообщение (заданное в виде строки) в массив байт в кодировке UTF-8. Вычисляем для полученного массива байт значение хэш функции (контрольной суммы).
3. Если вычисленное значение хэш функции (контрольной суммы) по модулю q равно 0, то присваиваем ему значение 1.
4. Вычисляем значение $v = h^{q-2} mod\, q$.
5. Вычисляем значения $z1 = (s * v)\, mod\, q$, $z2 = (q - r1) * v\, mod\, q$.
6. Вычисляем значение $u = (a^{z1} * y^{z2} mod\, p)\, mod\, q$.
7. Проверяем равенство u = r1. Если равенство выполняется, то подпись принимается. В противном подпись считается недействительной.

Основная программа:
1. Генерируем ЭЦП для исходного сообщения
2. Проверяем сгенерированное ЭЦП. Проверка должна пройти успешно: подпись принимается.
3. Меняем сгенерированное ЭЦП
4. Проверяем измененное ЭЦП. Проверка должна пройти не успешно: подпись недействительна.

Тексты программы:

В качестве языка программирования я использую C#. Исходный текст программы приведен в отдельном файле Program.cs (без дополнительных файлов проекта - *.csproj и решения - *.sln).

Результаты работы программы:
ЭЦП: r1 = 851, s = 1002
Сгенерированная ЭЦП проверку прошла успешно: подпись принимается.
Измененная ЭЦП проверку прошла неуспешно: подпись недействительна.



Анализ результатов:

1. Алгоритм выработки и проверки электронной цифровой подписи (ЭЦП) позволяет нам сопоставлять исходное сообщение и его подпись, т.е. мы всегда можем проверить соответствует ли исходному сообщению его ЭЦП и наоборот. Это позволяет поддерживать целостность исходных данных. Также мы можем гарантировать, что исходное сообщение, если оно было послано вместе с ЭЦП, создано определенным лицом, владеющим закрытым ключом; т.е. мы можем подтвердить авторство исходного сообщения.

2. Использование криптографически стойких функций для вычисления значения хеша крайне важно. В противном случае мы можем получить ситуацию, когда хеш функция, например, не стойка к коллизиям первого или второго рода. В этом примере мы использовали хеш функцию (на самом деле контрольную сумму), которая не стойка к коллизиям и первого и второго рода. Так например, у букв A (ASCII код 0b01000001) и B (ASCII код 0b01000010) значение хеш функции (количество 1 в битовом представлении символов) будет одинаково - 2; это означает, что у строк AA и BB будет одинаковое значение хеш функции и, как результат, создается одинаковая ЭЦП.
3. Алгоритм выработки и проверки электронной цифровой подписи (ЭЦП) очень чувствителен к качеству реализации некоторых алгоритмов; так например, если быстрое возведение в степень заменить на обычное, то скорость работы алгоритма выработки и проверки электронной цифровой подписи (ЭЦП) упадет в разы (у быстрого возведения в степень сложность O(logN), у обычного - O(N)).

Выводы:

Мы изучили и реализовали алгоритм для выработки и проверки электронной цифровой подписи (ЭЦП); проверили, что пара исходное сообщение и его ЭЦП позволяют удостоверится в целостности исходных данных.

Контрольные вопросы:

1. Какие криптоалгоритмы используются для создания электронной цифровой подписи?
Существует несколько схем построения электронной цифровой подписи:
1. На основе алгоритмов симметричного шифрования. Данная схема предусматривает наличие в системе третьего лица — арбитра, пользующегося доверием обеих сторон. Авторизацией документа является сам факт зашифрования его секретным ключом и передача его арбитру.
2. На основе алгоритмов асимметричного шифрования. На данный момент такие схемы ЭЦП наиболее распространены и находят широкое применение.
Мы в основном используем ЭЦП на основе алгоритмов асимметричного шифрования. Для того, чтобы использование цифровой подписи имело смысл, необходимо выполнение двух условий:
1. Верификация подписи должна производиться открытым ключом, соответствующим именно тому закрытому ключу, который использовался при подписании.
2. Без обладания закрытым ключом должно быть вычислительно сложно создать легитимную цифровую подпись.
Как уже было сказано, чтобы применение ЭП имело смысл, необходимо, чтобы вычисление легитимной подписи без знания закрытого ключа было вычислительно сложным процессом. Обеспечение этого во всех асимметричных алгоритмах цифровой подписи опирается на следующие вычислительные задачи:
1. Задачу дискретного логарифмирования (EGSA).
2. Задачу факторизации, то есть разложения числа на простые множители (RSA)

2. Что такое криптографическая хэш-функция, какими свойствами она должна обладать?

Криптографические хеш-функции — это выделенный класс хеш-функций (), который имеет определенные свойства, делающие его пригодным для использования в криптографии.
К криптографическим хеш-функциям предъявляются следующие требования:
1. **Сопротивление поиску прообраза**: при наличии хеша h должно быть трудно найти какое-либо сообщение m, такое что h = hash(m). Это свойство связано с понятием односторонней функции. Функции, у которых отсутствует это свойство, уязвимы для атак нахождения первого прообраза.
2. **Сопротивление поиску второго прообраза**: при наличии сообщения m_1, должно быть трудно найти другое сообщение m_2 (не равное m_1) такое, что $hash(m_1) = hash(m_2)$. Это свойство иногда называют слабым сопротивлением поиску коллизий. Функции, у которых отсутствует это свойство, уязвимы для атак поиска второго прообраза.
3. **Стойкость к коллизиям**: Коллизией для хеш-функции называется такая пара значений m и m′, m ≠ m′, для которой hash(m) = hash(m′). Так как количество возможных открытых текстов больше числа возможных значений свертки, то для некоторой свертки найдётся много прообразов, а следовательно, коллизии для хеш-функций обязательно существуют. Например, пусть длина хеш-прообраза 6 битов, длина свёртки 4 бита. Тогда число различных свёрток — 2^4 = 16, а число хеш-прообразов — 2^6 = 64, то есть в 4 раза больше, значит хотя бы одна свертка из всех соответствует 4 прообразам. Стойкость хеш-функции к коллизиям означает, что нет эффективного полиномиального алгоритма, позволяющего находить коллизии.

3. Как содержание сообщение влияет на электронную цифровую подпись?
ЭЦП создается на базе значения хеш функции от содержания сообщения. Изменение сообщения приводит к изменению значения хеш функции (если у нас не случилось коллизии) и, следовательно, к изменению значению ЭЦП.

4. Где используется ЭЦП?
Электронная подпись (ЭЦП) используется для идентификации отправителя документа, ускорения документооборота, а также для подтверждения неизменности и достоверности подписанной информации. Согласно действующему законодательству № 63-ФЗ от 06.04.2011, усиленная квалифицированная электронная подпись полностью идентична собственноручной подписи и оттиску печати на бумажном документе, в то время как простая и усиленная неквалифицированная электронная подпись могут быть идентичны собственноручной лишь при соблюдении определенных условий. Использование электронной подписи упрощает физическим лицам подачу документов в государственные органы и оптимизирует внутренний и внешний документооборот компаний.

5. В каком случае электронная цифровая подпись при проверке отвергается?
Есть проверке ЭЦА, она может быть отвергнута в двух случаях (см. алгоритм):
1. Если нарушено хотя бы одно из условий: 0 < r1 < q, 0 < s < q
2. Если не выполняется равенство u = r1
Оба эти случая означают, что ЭЦП и сообщение не соответствуют друг другу: либо были внесены изменения в сообщение, либо в ЭЦП.

6. От каких угроз информации защищает ЭЦП?
ЭЦП обеспечивает:

1. Защиту от изменений документа. При любом случайном или преднамеренном изменении документа (или подписи) подпись станет недействительной, потому что вычислена она на основании исходного состояния документа и соответствует лишь ему.
2. Невозможность отказа от авторства. Так как создать корректную подпись можно лишь, зная закрытый ключ, а он известен только владельцу, то владелец не может отказаться от своей подписи под документом.
От этих угроз и защищает наличие ЭЦП.