4. ЛАБОРАТОРНАЯ РАБОТА № 4. ЭЛЕКТРОННО-ЦИФРОВАЯ ПОДПИСЬ И ПРИЕМЫ ХЕШИРОВАНИЯ

Цель: овладеть практическими навыками закрытия информации электронно-цифровой подписью и приемами хеширования, рассмотрение хеширования методом контрольных сумм и методом наложения кодов — гаммированием.

4.1. Ход работы

1. Ознакомиться с теоретической частью данной работы.
2. Составить программу шифрования методом контрольных сумм.
3. Составить программу шифрования методом хеширования с применением гаммирования.
4. Составить отчет по проделанной работе.
5. Защитить работу.

4.2. Содержание отчета

1. Тема.
2. Цель работы.
3. Ход работы.
4. Постановка задачи.
5. Листинг программы.
6. Результат выполнения программы.

4.3. Теоретическая справка

Любая подпись (или иной способ подтверждения подлинности документа), будь то обычная или электронная, всегда выполняет, по крайней мере, три функции:

1) функцию авторизации — подтверждение того, что подписавшийся действительно является тем, за кого мы его принимаем;

2) обеспечение того, что подписавшийся не может отказаться от документа, который он подписал;

3) подтверждение того, что отправитель подписал именно тот документ, который отправил, а не какой-либо иной.

Первые две функции обеспечивают защиту лица, для которого документ предназначен (адресат), а третья — интересы подписавшегося (корреспондента). Во всех случаях проявляется свойство подписи, называемое аутентичностью (подлинностью). Свойство аутентичности подписи переносится на весь документ в целом.

При выработке электронной цифровой подписи (ЭЦП) и ее расшифровании получателем корреспондент и адресат пользуются методом несимметричного шифрования.

В упрощенном виде ЭЦП формируется следующим образом.

1. Корреспондент *X* по специальному алгоритму обрабатывает документ, предназначенный для отправки адресату *Y*. В результате применения этого алгоритма, вырабатывается некоторый параметр, характеризующий документ в целом. Объем памяти, занимаемый выработанным параметром, значительно меньше, чем объем всего документа (1, 2, 4 байта).

2. Затем *X* с помощью секретной части ключа шифрует полученный параметр. Полученный таким образом шифр является ЭЦП корреспондента *X*.

3. Корреспондент *X* отправляет адресату *Y* документ и свою электронную цифровую подпись.

4. Адресат *Y* реализует на полученном документе тот же алгоритм, которым пользовался корреспондент *X*.

5. Затем *Y* дешифрует электронную цифровую подпись, полученную от *X*, пользуясь открытой частью ключа, предоставленной ему корреспондентом *X*.

6. Окончательно адресат *Y* сравнивает значение параметра, полученного на четвертом этапе, с расшифрованным значением ЭЦП. Если эти значения совпадают, то подпись подлинная и документ при передаче не был изменен. В противном случае — либо документ искажен, либо подпись подделана, либо и то и другое.

Применение ЭЦП не предполагает обязательного засекречивания (шифрования) самого передаваемого документа. Шифруется только некоторая интегральная характеристика этого документа. Если документ при передаче по каналу связи будет изменен злоумышленником, то, естественно, изменится и его интегральная характеристика, а это сразу заметит адресат при расшифровании ЭЦП. В связи с этим встает вопрос, каким должен быть алгоритм получения интегральной характеристики документа (не слишком большого параметра, характеризующего документ в целом).

Значение интегрального параметра называют хеш-значением документа, а способ (алгоритм) получения хеш-значения — хеш-функцией. Получение хеш-значения с помощью хеш-функции называют сворачиванием (хешированием) текста документа в более короткий текст (интегральный параметр).

Обратимся к важному вопросу хеширования данных. Предположим, что имеется некоторый текст *P* — последовательность знаков

некоторого алфавита — и некоторый алгоритм A, преобразующий P в некоторый текст M меньшей длины:

$$M = A(P). \quad (4)$$

Ясно, что алгоритм хеширования A должен быть таким, чтобы при случайном равновероятном выборе двух текстов P_1 и P_2, из множества возможных, соответствующие тексты M_1 и M_2 с высокой вероятностью были бы различны. Поскольку текст P длиннее (содержит значительно большее количество двоичных разрядов при двоичном кодировании) хеш-значения M, то, вообще говоря, существует много текстов P с одним и тем же хеш-значением M. Однако алгоритм A организуется так, что невозможно однозначно по хеш-значению M восстановить сам текст. В этом проявляется отсутствие свойства взаимной однозначности между множеством исходных текстов $\{P\}$ и множеством хеш-значений $\{M\}$. Кроме того, сложность алгоритма A должна обеспечить невозможность осмысленного изменения текста P с сохранением того же самого хеш-значения M.

Рассмотрим некоторые способы хеширования.

Метод контрольных сумм

Исторически это самый первый и самый простой способ хеширования, который использовался для проверки правильности ввода программ и данных еще в ЭВМ первого поколения.

Под контрольной суммой понимается некоторое значение, рассчитанное путем сложения всех чисел (кодов символов), соответствующих данному тексту. Если сумма всех таких чисел K превышает максимально допустимое значение ($\mathrm{Max}Val$), заданное заранее, то величина контрольной суммы полагается равной остатку от деления полученной суммы на максимально возможное значение контрольной суммы, увеличенное на единицу. Таким образом, контрольную сумму можно записать в следующем виде:

$$K\mathrm{Summ} = \begin{cases} K, & \text{при } K \le \mathrm{Max}Val; \\ K \bmod(\mathrm{Max}Val + 1), & \text{при } K > \mathrm{Max}Val. \end{cases} \quad (5)$$

Пример. Допустим, что документ, который следует подписать ЭЦП, представляет собой следующий текст из романа Ф. М. Достоевского «Идиот»:

«Смиренный игумен Пафнутий руку приложил».

В соответствии с системой кодирования ASCII написанное предложение (вместе с последней точкой) представляет собой последовательность целых чисел, записанных в десятичной системе счисления:

145 172 168 224 165 173 173 235 169 32 168 163 227 172 165 173
32 143 160 228 173 227 226 168 169 32 224 227 170 227
32 175 224 168 171 174 166 168 171 46.

Сумма всех кодов K = 6625. Задав значения MaxVal = 3776, получаем KSumm = 2848 (вычислили остаток от деления K на 3777).

Затем полученную контрольную сумму KSumm = 2848 шифруем с помощью открытой части ключа и посылаем адресату.

Если весь текст необходимо сжать (хешировать) в параметр длиной в один байт, то можно в качестве KSumm взять остаток от деления K на 256. В приведенном примере при этом получаем KSumm = 225.

Контрольную сумму можно вычислить и по-другому. Представим все коды символов документа в виде двоичных слов. Каждое такое слово имеет длину 8 битов (1 байт). Например, в приведенной фразе (цитата из Ф. М. Достоевского) символы кодируются двоичными словами:

10010001 — 145 10101100 — 172 10101000 — 168

……

10101011 — 171 00101110 — 46

Контрольную сумму можно составить как поразрядную сумму по модулю 2 ($\oplus$) всех двоичных кодов текста P. В нашем примере получается: KSumm = 10011011.

Метод контрольных сумм впервые был применен для тестирования правильности ввода данных в ЭВМ, т. е. для контроля работы технических устройств. Поскольку сбои в работе устройств ЭВМ нецеленаправленны (случайны), метод контрольных сумм давал надежный результат. Иная ситуация в случае с ЭЦП. Человек (злоумышленник) будет стараться изменить документ в свою пользу так, чтобы контрольная сумма не изменялась.

Недостаток метода контрольных сумм (в обоих вариантах) заключается в том, что хотя несовпадение значений этих сумм служит верным признаком того, что документ подвергся изменению, но равенство значений еще не дает гарантии, что информация осталась неизменной. Можно произвольным образом изменить порядок следования букв, цифр или слов и фраз в документе, при этом контрольная сумма сохранит прежнее значение. Так предложениям «казнить нельзя, помиловать» и «казнить, нельзя помиловать» соответствуют одни и те же контрольные суммы, а их содержание прямо противоположное. И что еще хуже — можно изменить отдельные числа в документе и подогнать остальные так, что контрольная сумма останется той же самой. Например, вместо суммы в 1 000 005 рублей написать 1 500 000 рублей и получить ни за что половину миллиона рублей.

Внесение небольших изменений в получение контрольной суммы

Этот метод позволяет преодолеть названные недостатки. Изменение состоит в том, что, прежде чем вычислять контрольную сумму, на каждый код текста накладывается специальный код. Совокупность этих кодов в теории шифрования носит название гаммы шифра, метод наложения кодов — гаммирование. Опишем этот метод.

Пусть каждому символу документа (открытого текста) соответствует восьмибитовое двоичное слово X_i. Таким образом, исходный документ представляется в виде последовательности восьмибитовых двоичных слов: $X_1, X_2, \ldots, X_p$.

Затем выработаем последовательность псевдослучайных чисел t_i по рекуррентной формуле

$$t_{i+1} = (a \cdot t_i + b) \bmod c, \qquad (6)$$

где $i = 0, 1, \ldots, p - 1$; a, b, t_0 — заданные числа; p — количество символов в тексте.

При $c = 2^n$. Если взять $n = 8$, то двоичные представления чисел t_i не будут превышать восьми двоичных знаков.

Далее каждое число t_i представим в виде восьмибитового двоичного слова. Получаем последовательность двоичных слов: $T_1, T_2, \ldots, T_p$.

Двоичные числа X_i и T_i, сложим поразрядно по модулю 2. Получим новую последовательность двоичных слов:

$$Y_1 = X_1 \oplus T_1, Y_2 = X_2 \oplus T_2, \ldots, Y_p = X_p \oplus T_p.$$

Каждое двоичное слово, рассматриваемое как двоичное число, переведем в десятичную систему, при этом получим последовательность чисел: $y_1, y_2, \ldots, y_p$.

Полученная последовательность целых чисел суммируется по модулю MaxVal + 1 (если $n = 8$, то MaxVal = 255).

4.4. Задание к лабораторной работе № 4

Составить программу шифрования методом контрольных сумм и методом хеширования с применением гаммирования.

Варианты заданий к лабораторной работе № 4

Вариант 1

Пусть $a = 17$, $b = 11$, c = MaxVal + 1 = 256, $t_0 = 172$. Вычислить контрольные суммы для нескольких сообщений методом контрольных сумм (KSumm) и методом хеширования с применением гаммирования (SummKodBukvOtkr):

а) P = '0123456789', KSumm = ?, SummKodBukvOtkr – ?;

б) P = '9876543210', KSumm = ?, SummKodBukvOtkr – ?;

в) P = '1000005', KSumm = ?, SummKodBukvOtkr – ?;
г) P = '1500000', KSumm = ?, SummKodBukvOtkr – ?

Вариант 2

Пусть a = 13, b = 19, c = MaxVal + 1 = 256, t_0 = 155. Вычислить контрольные суммы для нескольких сообщений методом контрольных сумм (KSumm) и методом хеширования с применением гаммирования (SummKodBukvOtkr):

а) P = '0123456789', KSumm = ?, SummKodBukvOtkr – ?;
б) P =' 9876543210', KSumm = ?, SummKodBukvOtkr – ?;
в) P = ' 1000005', KSumm = ?, SummKodBukvOtkr – ?;
г) P = ' 1500000', KSumm = ?, SummKodBukvOtkr –?

Вариант 3

Пусть a = 23, b = 7, c = MaxVal + 1 = 256, t_0 = 131. Вычислить контрольные суммы для нескольких сообщений методом контрольных сумм (KSumm) и методом хеширования с применением гаммирования (SummKodBukvOtkr):

а) P = '0123456789', KSumm = ?, SummKodBukvOtkr – ?;
б) P =' 9876543210', KSumm = ?, SummKodBukvOtkr – ?;
в) P = ' 1000005', KSumm = ?, SummKodBukvOtkr – ?;
г) P = ' 1500000', KSumm = ?, SummKodBukvOtkr – ?

Вариант 4

Пусть a = 19, b = 3, c = MaxVal + 1 = 256, t_0 = 101. Вычислить контрольные суммы для нескольких сообщений методом контрольных сумм (KSumm) и методом хеширования с применением гаммирования (SummKodBukvOtkr):

а) P = '02468', KSumm = ?, SummKodBukvOtkr – ?;
б) P =' 86420', KSumm = ?, SummKodBukvOtkr – ?;
в) P = '1000009', KSumm = ?, SummKodBukvOtkr – ?;
г) P = '1900000', KSumm = ?, SummKodBukvOtkr – ?

Вариант 5

Пусть a = 17, b = 3, c = MaxVal + 1 = 256, t_0 = 191. Вычислить контрольные суммы для нескольких сообщений методом контрольных сумм (KSumm) и методом хеширования с применением гаммирования (SummKodBukvOtkr):

а) P = '013579 ', KSumm = ?, SummKodBukvOtkr – ?;
б) P ='975310 ', KSumm = ?, SummKodBukvOtkr – ?;
в) P = '1000006 ', KSumm = ?, SummKodBukvOtkr – ?;
г) P = '1600000 ', KSumm = ?, SummKodBukvOtkr – ?

Вариант 6

Пусть a = 31, b = 5, c = Max*Val* + 1 = 256, t_0 = 121. Вычислить контрольные суммы для нескольких сообщений методом контрольных сумм (*K*Summ) и методом хеширования с применением гаммирования (SummKodBukvOtkr):

а) P = '001133557799', *K*Summ = ?, SummKodBukvOtkr – ?;

б) P = '997755331100', *K*Summ = ?, SummKodBukvOtkr – ?;

в) P = '1000008', *K*Summ = ?, SummKodBukvOtkr – ?;

г) P = '1800000', *K*Summ = ?, SummKodBukvOtkr – ?

Вариант 7

Пусть a = 37, b = 11, c = Max*Val* + 1 = 256, t_0 = 221. Вычислить контрольные суммы для нескольких сообщений методом контрольных сумм (*K*Summ) и методом хеширования с применением гаммирования (SummKodBukvOtkr):

а) P = '021135579', *K*Summ = ?, SummKodBukvOtkr – ?;

б) P = '975531120', *K*Summ = ?, SummKodBukvOtkr – ?;

в) P = '1000097', *K*Summ = ?, SummKodBukvOtkr – ?;

г) P = '1970000', *K*Summ = ?, SummKodBukvOtkr – ?

Вариант 8

Пусть a = 9, b = 11, c = Max*Val* + 1 = 256, t_0 = 201. Вычислить контрольные суммы для нескольких сообщений методом контрольных сумм (*K*Summ) и методом хеширования с применением гаммирования (SummKodBukvOtkr):

а) P = '021345', *K*Summ = ?, SummKodBukvOtkr – ?;

б) P = '543120', *K*Summ = ?, SummKodBukvOtkr – ?;

в) P = '1000999', *K*Summ = ?, SummKodBukvOtkr – ?;

г) P = '1999000', *K*Summ = ?, SummKodBukvOtkr – ?

Вариант 9

Пусть a = 23, b = 19, c = Max*Val* + 1 = 256, t_0 = 235. Вычислить контрольные суммы для нескольких сообщений методом контрольных сумм (*K*Summ) и методом хеширования с применением гаммирования (SummKodBukvOtkr):

а) P = '0000123456', *K*Summ = ?, SummKodBukvOtkr – ?;

б) P = '6543210000', *K*Summ = ?, SummKodBukvOtkr – ?;

в) P = '10000001', *K*Summ = ?, SummKodBukvOtkr – ?; г) P = '11000000', *K*Summ = ?, SummKodBukvOtkr – ?

Вариант 10

Пусть a = 31, b = 7, c = Max*Val* + 1 = 256, t_0 = 126. Вычислить контрольные суммы для нескольких сообщений методом контроль-

ных сумм (*K*Summ) и методом хеширования с применением гаммирования (SummKodBukvOtkr):

а) *P* = '00009999', *K*Summ = ?, SummKodBukvOtkr – ?;

б) *P* = '99990000', *K*Summ = ?, SummKodBukvOtkr – ?;

в) *P* = '10000001', *K*Summ = ?, SummKodBukvOtkr – ?;

г) *P* = '11000000', *K*Summ = ?, SummKodBukvOtkr – ?

Вариант 11

Пусть *a* = 41, *b* = 9, *c* = Max*Val* + 1 = 256, t_0 = 192. Вычислить контрольные суммы для нескольких сообщений методом контрольных сумм (*K*Summ) и методом хеширования с применением гаммирования (SummKodBukvOtkr):

а) *P* = '11115555', *K*Summ = ?, SummKodBukvOtkr – ?;

б) *P* = '55551111', *K*Summ = ?, SummKodBukvOtkr – ?;

в) *P* = '10000001', *K*Summ = ?, SummKodBukvOtkr – ?;

г) *P* = '11000000', *K*Summ = ?, SummKodBukvOtkr – ?

Вариант 12

Пусть *a* = 51, *b* = 13, *c* = Max*Val* + 1 = 256, t_0 = 102. Вычислить контрольные суммы для нескольких сообщений методом контрольных сумм (*K*Summ) и методом хеширования с применением гаммирования (SummKodBukvOtkr):

а) *P* = '12121212', *K*Summ = ?, SummKodBukvOtkr – ?;

б) *P* = '21212121', *K*Summ = ?, SummKodBukvOtkr – ?;

в) *P* = '90000009', *K*Summ = ?, SummKodBukvOtkr – ?;

г) *P* = '99000000', *K*Summ = ?, SummKodBukvOtkr – ?

Вариант 13

Пусть *a* = 61, *b* = 5, *c* = Max*Val* + 1 = 256, t_0 = 212. Вычислить контрольные суммы для нескольких сообщений методом контрольных сумм (*K*Summ) и методом хеширования с применением гаммирования (SummKodBukvOtkr):

а) *P* = '191919', *K*Summ = ?, SummKodBukvOtkr – ?;

б) *P* = '919191', *K*Summ = ?, SummKodBukvOtkr – ?;

в) *P* = '10000009', *K*Summ = ?, SummKodBukvOtkr – ?;

г) *P* = '19000000', *K*Summ = ?, SummKodBukvOtkr – ?

Вариант 14

Пусть *a* = 71, *b* = 13, *c* = Max*Val* + 1 = 256, t_0 = 144. Вычислить контрольные суммы для нескольких сообщений методом контрольных сумм (*K*Summ) и методом хеширования с применением гаммирования (SummKodBukvOtkr):

а) P = '100009', KSumm = ?, SummKodBukvOtkr – ?;
б) P = '900001', KSumm = ?, SummKodBukvOtkr – ?;
в) P = '100009', KSumm = ?, SummKodBukvOtkr – ?;
г) P = '190000', KSumm = ?, SummKodBukvOtkr – ?

Вариант 15

Пусть a = 17, b = 7, c = MaxVal + 1 = 256, t_0 = 152. Вычислить контрольные суммы для нескольких сообщений методом контрольных сумм (KSumm) и методом хеширования с применением гаммирования (SummKodBukvOtkr):

а) P = '100009', KSumm = ?, SummKodBukvOtkr – ?;
б) P = '900001', KSumm = ?, SummKodBukvOtkr – ?;
в) P = '100009', KSumm = ?, SummKodBukvOtkr – ?;
г) P = '190000', KSumm = ?, SummKodBukvOtkr – ?

Контрольные вопросы

1. Назвать три функции ЭЦП.
2. Перечислить этапы формирования ЭЦП.
3. Что шифруется при применении ЭЦП?
4. Что называется хеш-значением документа?
5. Что называется хеш-функцией?
6. Что называется сворачиванием (хешированием) документа?
7. В чем заключается метод контрольных сумм?
8. Перечислить этапы метода хеширования с применением гаммирования.
9. Недостаток метода контрольных сумм.