При электронном документообороте традиционные методы подтверждения подлинности документа при помощи оттиска печати или поставленной от руки подписи являются непригодными. Вместо этого используется электронная цифровая подпись (ЭЦП) под электронным документом, представляющая собой небольшой объём данных, передаваемых вместе с документом при его пересылке.

Любая система электронной цифровой подписи включает в себя процедуру подписания электронного документа и процедуру проверки данной подписи.

Наибольшим удобством использования обладают схемы цифровой подписи на основе криптосистем с открытым ключом. Для таких схем при постановке подписи используется закрытый ключ автора электронного документа, а при проверке подписи – открытый ключ.

Важной составляющей любой системы электронной цифровой подписи является процедура вычисления дайджеста сообщения, необходимого для защиты передаваемой информации от случайного либо преднамеренного искажения. Для вычисления дайджеста используется криптографическая хеш-функция, формирующая хеш-образ отправляемого сообщения.

Алгоритм безопасного хеширования SHA-1

Алгоритм безопасного хеширования SHA-1 был опубликован в 1995 году в качестве замены использовавшегося до этого алгоритма хеширования SHA-0, в котором была обнаружена уязвимость.

Для сообщения произвольной длины l, не превышающей 2^{64} бит, алгоритм SHA-1 формирует 160-битный хеш-образ.

Процедура формирования хеш-образа состоит из следующих шагов.

1. Весь исходный текст разбивается на блоки по 512 бит. В случае если длина исходного текста не кратна 512 битам, производится его выравнивание за счёт добавления в конец бита со значением 1, m нулевых битов и 64-битного представления значения длины исходного сообщения (рис. 12).

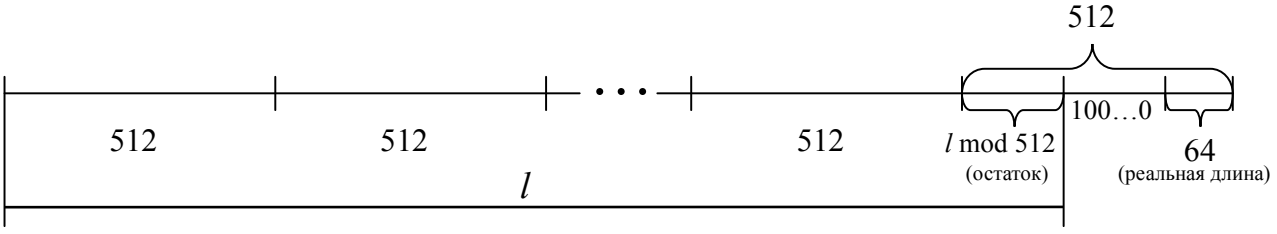

Рис. 12. Выравнивание исходного сообщения

2. Инициализируется пять 32-битных рабочих переменных A, B, C, D, E:

$A \leftarrow$ 67452301$_{16}$

$B \leftarrow$ EFCDAB89$_{16}$

$C \leftarrow$ 98BADCFE$_{16}$

$D \leftarrow$ 10325476$_{16}$

$E \leftarrow$ C3D2E1F0$_{16}$

3. Выполняется обработка очередных 512 бит исходного текста. Для этого значения переменных A, B, C, D, E копируются в переменные a, b, c, d, e и далее для t от 1 до 80 выполняется преобразование значений данных переменных по схеме, изображенной на рис. 13.

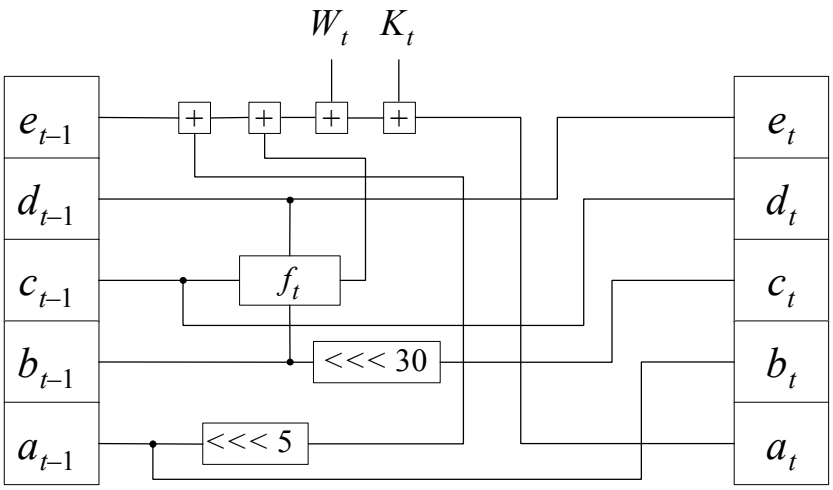

Рис. 13. Схема итерации алгоритма SHA-1

Каждая из 80 итераций может быть записана следующим образом:

$TMP \leftarrow (a <<< 5) + f_t(b, c, d) + e + W_t + K_t;$
$e \leftarrow d;$
$d \leftarrow c;$
$c \leftarrow (b <<< 30);$
$b \leftarrow a;$
$a \leftarrow TMP,$

где «+» – операция сложения по модулю 2^{32}, $f_t(X, Y, Z)$ – нелинейная функция, имеющая следующий вид:

$$f_t(X, Y, Z) = \begin{cases} (X \,\&\, Y) \mid (!X \,\&\, Z), & t \in \overline{1, 20}, \\ X \oplus Y \oplus Z, & t \in \overline{21, 40}, \\ (X \,\&\, Y) \mid (X \,\&\, Z) \mid (Y \,\&\, Z), & t \in \overline{41, 60}, \\ X \oplus Y \oplus Z, & t \in \overline{61,80}, \end{cases} \quad (22)$$

где «&» – побитовая операция «И», «|» – побитовая операция «ИЛИ», «!» – операция побитового инвертирования, « $\oplus$ » – операция побитового сложения по модулю 2. Параметр K_t принимает четыре различных значения в зависимости от номера текущей итерации:

$K_t = \texttt{5A827999}_{16}$, $t \in \overline{1,20}$;

$K_t = \texttt{6ED9EBA1}_{16}$, $t \in \overline{21,40}$;

$K_t = \texttt{8F1BBCDC}_{16}$, $t \in \overline{41,60}$;

$K_t = \texttt{CA62C1D6}_{16}$, $t \in \overline{61,80}$.

«<<<» – операция циклического сдвига на 30 либо 5 бит влево, W_t – одно из шестнадцати 32-битных слов 512-битного блока сообщения при $t \in \overline{1,16}$ либо значение, определяемое в соответствии со следующим выражением при $t \in \overline{17,80}$:

$$W_t = (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) <<< 1. \quad (23)$$

4. Значения переменных a, b, c, d, e независимо друг от друга складываются по модулю 2^{32} со значениями переменных A, B, C, D, E, в которые затем и помещаются полученные результаты.

5. Шаги 3–4 выполняются до тех пор, пока не будет обработан весь текст.

После обработки последнего блока текста значение хеш-образа формируется как $ABCDE$.

Алгоритм цифровой подписи RSA

Математическая схема электронной цифровой подписи по алгоритму RSA была предложена в 1977 году сотрудниками Массачусетского технологического института США. Данная система цифровой подписи стала первым практическим решением задачи подписи электронных документов при помощи криптосистем с открытым ключом. Процедура вычисления цифровой подписи в данной системе использует криптографическое преобразование по алгоритму RSA.

В соответствии с данной системой цифровой подписи, субъект, желающий пересылать подписанные им документы, должен сформировать два ключа алгоритма RSA: открытый и закрытый.

Пару значений (K_O, r), которая является открытым ключом подписи, отправитель передаёт всем возможным получателям его сообщений. Именно эти значения будут использоваться для проверки подлинности и принадлежности отправителю полученных от него сообщений.

Значение K_C сохраняется отправителем в секрете. Данное значение вместе с модулем r является секретным ключом, который будет использоваться отправителем для постановки подписей под своими сообщениями.

Схема использования алгоритма цифровой подписи на базе RSA для обмена двух абонентов подписанными сообщениями показана на рис. 14.

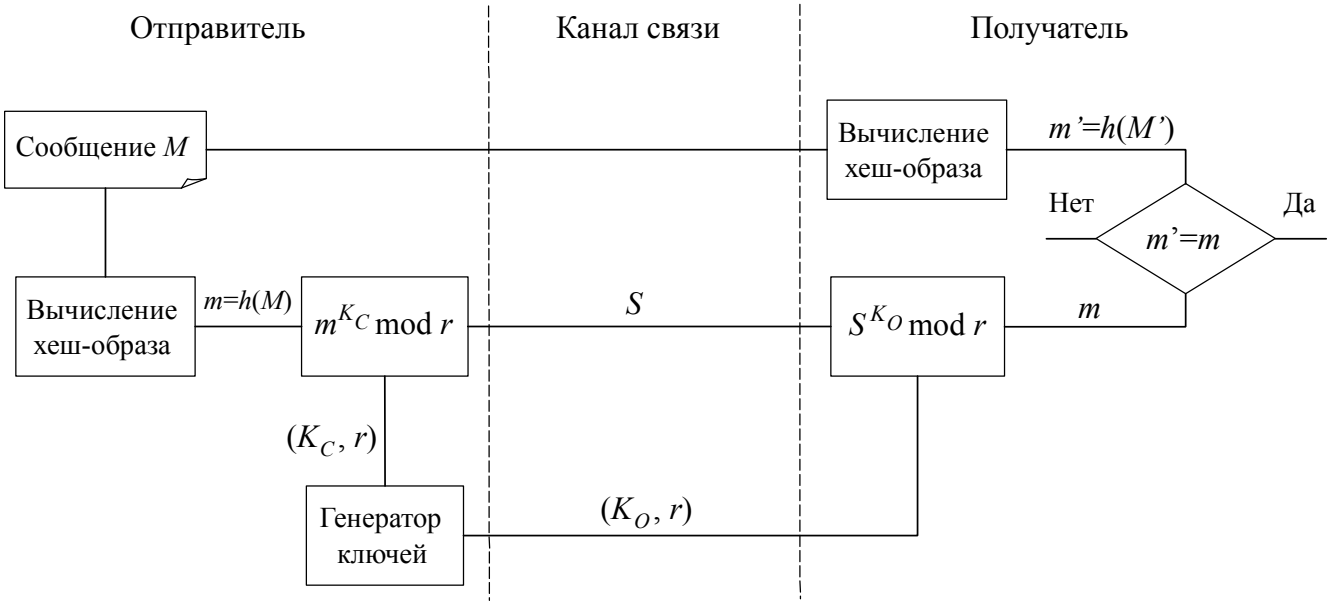

Рис. 14. Схема использования цифровой подписи на базе RSA

Допустим, что получатель уже располагает открытым ключом подписи отправителя. Процедура подписи отправителем сообщения *M* будет состоять из следующих шагов.

1. Отправитель сжимает сообщение *M* при помощи криптографической хеш-функции *h* в целое число $m = h(M)$.

2. Отправитель вычисляет значение цифровой подписи *S* для сообщения *M* на основе ранее полученного значения хеш-образа *m* и значения своего закрытого (секретного) ключа подписи K_C. Для этого используется преобразование, аналогичное преобразованию, выполняемому при шифровании по алгоритму RSA:

$$S = m^{KC} \bmod r. \tag{24}$$

Пара (*M*, *S*), представляющая собой подписанное отправителем сообщение, передаётся получателю. Сформировать подпись *S* мог только обладатель закрытого ключа K_C.

Процедура проверки получателем подлинности сообщения и принадлежности его отправителю состоит из следующих шагов.

1. Получатель сжимает полученное сообщение *M'* при помощи криптографической хеш-функции *h*, идентичной той, которая была использована отправителем, в целое число *m'*.

2. Получатель выполняет расшифрование открытым ключом K_O отправителя дайджеста *m* оригинального сообщения, преобразуя значение подписи *S* по алгоритму RSA:

$$m = S^{KO} \bmod r. \tag{25}$$

3. Получатель сравнивает полученные значения *m'* и *m*. Если данные значения совпадают, т. е.

$$S^{KO} \bmod r = h(M'), \tag{26}$$

то получатель признает полученное сообщение подлинным и принадлежащим отправителю.

Фальсификация сообщения при его передаче по каналу связи возможна только при получении злоумышленником секретного ключа K_C либо за счет проведения успешной атаки против использованной для вычисления дайджеста сообщения хеш-функции. При использовании достаточно больших значений p и q определение секретного значения K_C по открытому ключу (K_O, r) является чрезвычайно трудной задачей, соответствующей по сложности разложению модуля r на множители. Используемые в реальных приложениях хеш-функции обладают характеристиками, делающими атаку против цифровой подписи практически неосуществимой.

Задания

1. Реализовать алгоритм вычисления хеш-функции SHA-1 для файла с произвольным размером и содержимым.
2. Реализовать программное средство, выполняющее генерацию и проверку ЭЦП файла с произвольным содержимым на базе алгоритма RSA с использованием для вычисления хеш-функции ранее реализованного алгоритма SHA-1.