Практическая работа № 1. Электронная цифровая подпись

1. Цель работы

Обзор и изучение принципов формирования электронной цифровой подписи.

Практическая работа № 1. Электронная цифровая подпись

1. Цель работы

Практическая работа № 1. Электронная цифровая подпись

2. Основы теории

При обмене электронными документами по сети связи существенно снижаются затраты на обработку и хранение документов, убыстряется их поиск. Но при этом возникает проблема аутентификации автора документа и самого документа, т.е. установления подлинности автора и отсутствия изменений в полученном документе. В обычной (бумажной) информатике эти проблемы решаются за счет того, что информация в документе и рукописная подпись автора жестко связаны с физическим носителем (бумагой). В электронных документах на машинных носителях такой связи нет.

Целью аутентификации электронных документов является их защита от возможных видов злоумышленных действий, к которым относятся:

- активный перехват – нарушитель, подключившийся к сети, перехватывает документы (файлы) и изменяет их;
- маскарад – абонент C посылает документ абоненту B от имени абонента A;
- ренегатство – абонент A заявляет, что не посылал сообщения абоненту B, хотя на самом деле послал;
- подмена – абонент B изменяет или формирует новый документ и заявляет, что получил его от абонента A;
- повтор – абонент C повторяет ранее переданный документ, который абонент A посылал абоненту B.

Эти виды злоумышленных действий могут нанести существенный ущерб банковским и коммерческим структурам, государственным предприятиям и организациям, частным лицам, применяющим в своей деятельности компьютерные информационные технологии.

При обработке документов в электронной форме совершенно непригодны традиционные способы установления подлинности по рукописной подписи и оттиску печати на бумажном документе. Принципиально новым решением является электронная цифровая подпись (ЭЦП).

Электронная цифровая подпись используется для аутентификации текстов, передаваемых по телекоммуникационным каналам. Функционально она аналогична обычной рукописной подписи и обладает ее основными достоинствами:

- удостоверяет, что подписанный текст исходит от лица, поставившего подпись;
- не дает самому этому лицу возможности отказаться от обязательств, связанных с подписанным текстом;
- гарантирует целостность подписанного текста.

Цифровая подпись представляет собой относительно небольшое количество дополнительной цифровой информации, передаваемой вместе с подписываемым текстом.

Система ЭЦП включает две процедуры: 1) процедуру постановки подписи; 2) процедуру проверки подписи. В процедуре постановки подписи используется секретный ключ отправителя сообщения, в процедуре проверки подписи – открытый ключ отправителя.

При формировании ЭЦП отправитель прежде всего вычисляет хэш-функцию $h(M)$ подписываемого текста M. Вычисленное значение хэш-функции $h(M)$ представляет собой один короткий блок информации m, характеризующий весь текст M в целом. Затем число m шифруется секретным ключом отправителя. Получаемая при этом пара чисел представляет собой ЭЦП для данного текста M.

При проверке ЭЦП получатель сообщения снова вычисляет хэш-функцию $m = h(M)$ принятого по каналу текста M, после чего при помощи открытого ключа отправителя проверяет, соответствует ли полученная подпись вычисленному значению m хэш-функции.

Принципиальным моментом в системе ЭЦП является невозможность подделки ЭЦП пользователя без знания его секретного ключа подписывания.

В качестве подписываемого документа может быть использован любой файл. Подписанный файл создается из неподписанного путем добавления в него одной или более электронных подписей.

Каждая подпись содержит следующую информацию:

- дату подписи;
- срок окончания действия ключа данной подписи;
- информацию о лице, подписавшем файл (Ф.И.О., должность, краткое наименование фирмы);
- идентификатор подписавшего (имя открытого ключа);
- собственно цифровую подпись.

Однонаправленные хэш-функции

Хэш-функция предназначена для сжатия подписываемого документа M до нескольких десятков или сотен бит. Хэш-функция $h(\cdot)$ принимает в качестве аргумента сообщение (документ) M произвольной длины и возвращает хэш-значение $h(M)=H$ фиксированной длины. Обычно хешированная информация является сжатым двоичным представлением основного сообщения произвольной длины. Следует отметить, что значение хэш-функции $h(M)$ сложным образом зависит от документа M и не позволяет восстановить сам документ M.

Практическая работа № 1. Электронная цифровая подпись

2. Основы теории

Хэш-значение, вычисляемое при вводе последнего блока текста, становится хэш-значением всего сообщения M.

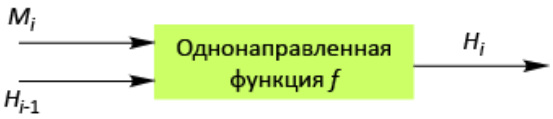

Рисунок 1 – Построение однонаправленной хэш-функции

В результате однонаправленная хэш-функция всегда формирует выход фиксированной длины n (независимо от длины входного текста).

Алгоритм безопасного хеширования SHA

Алгоритм безопасного хеширования SHA (Secure Hash Algorithm) разработан НИСТ и АНБ США в рамках стандарта безопасного хеширования SHS (Secure Hash Standard), в 1992 г. Алгоритм хеширования SHA предназначен для использования совместно с алгоритмом цифровой подписи DSA.

При вводе сообщения M произвольной длины менее 2^{64} бит алгоритм SHA вырабатывает 160-битовое выходное сообщение, называемое дайджестом сообщения MD (Message Digest). Затем этот дайджест сообщения используется в качестве входа алгоритма DSA, который вычисляет цифровую подпись сообщения M. Формирование цифровой подписи для дайджеста сообщения, а не для самого сообщения повышает эффективность процесса подписания, поскольку дайджест сообщения обычно намного короче самого сообщения.

Такой же дайджест сообщения должен вычисляться пользователем, проверяющим полученную подпись, при этом в качестве входа в алгоритм SHA используется полученное сообщение M.

Алгоритм хеширования SHA назван безопасным, потому что он спроектирован таким образом, чтобы было вычислительно невозможно восстановить сообщение, соответствующее данному дайджесту, а также найти два различных сообщения, которые дадут одинаковый дайджест. Любое изменение сообщения при передаче с очень большой вероятностью вызовет изменение дайджеста, и принятая цифровая подпись не пройдет проверку.

Рассмотрим подробнее работу алгоритма хеширования SHA. Прежде всего, исходное сообщение M дополняют так, чтобы оно стало кратным 512 битам. Дополнительная набивка сообщения выполняется следующим образом: сначала добавляется единица, затем следуют столько нулей, сколько необходимо для получения сообщения, которое на 64 бита короче, чем кратное 512, и, наконец, добавляют 64-битовое представление длины исходного сообщения.

Инициализируется пять 32-битовых переменных в виде:

$$A = 0x67452301$$
$$B = 0xEFCDAB89$$
$$C = 0x98BADCFE$$
$$D = 0x10325476$$
$$E = 0xC3D2E1F0$$

Затем начинается главный цикл алгоритма. В нем обрабатывается по 512 бит сообщения поочередно для всех 512-битовых блоков, имеющихся в сообщении. Первые пять переменных A, B, C, D, E копируются в другие переменные a, b, c, d, e: $a = A$, $b = B$, $c = C$, $d = D$, $e = E$.

Главный цикл содержит четыре цикла по 20 операций каждый. Каждая операция реализует нелинейную функцию от трех из пяти переменных a, b, c, d, e, а затем производит сдвиг и сложение.

Алгоритм SHA имеет следующий набор нелинейных функций:

$$f_t(X,Y,Z) = (X \wedge Y) \vee ((\neg X) \wedge Z) \quad \text{для } t = 0...19,$$
$$f_t(X,Y,Z) = (X \oplus Y \oplus Z) \quad \text{для } t = 20...39,$$
$$f_t(X,Y,Z) = (X \wedge Y) \vee (X \wedge Z) \vee (Y \wedge Z) \quad \text{для } t = 40...59,$$
$$f_t(X,Y,Z) = (X \oplus Y \oplus Z) \quad \text{для } t = 60...79,$$

где t – номер операции.

В алгоритме используются также четыре константы:

Практическая работа № 1. Электронная цифровая подпись

2. Основы теории

где t – номер операции (для t = 1 ...80),
W_t – t-й субблок расширенного сообщения,
<<<S – циклический сдвиг влево на S бит.

С учетом введенных обозначений главный цикл из восьмидесяти операций можно описать так:

$$FOR\ t\ = 0\ до\ 79$$
$$TEMP\ =\ (a\ <<<\ 5)\ +\ f_t(b,c,d)\ +\ e\ +\ W_t + K_t$$
$$e\ =\ d$$
$$d\ =\ c$$
$$c = (b\ <<< 30)$$
$$b\ =\ a$$
$$a\ =\ TEMP$$

Схема выполнения одной операции показана на рисунке 2.

После окончания главного цикла значения a, b, c, d и e складываются с A, B, C, D и E соответственно, и алгоритм приступает к обработке следующего 512-битового блока данных. Окончательный выход формируется в виде конкатенации значений A, B, C, D и E.

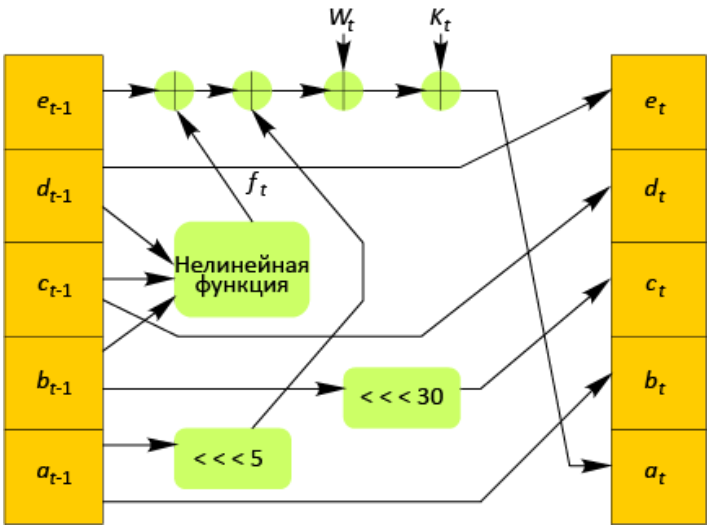

Рисунок 2 – Схема выполнения одной операции алгоритма SHA

Поскольку алгоритм SHA выдает 160-битовое хэш-значение, он более устойчив к атакам полного перебора и атакам "дня рождения", чем большинство других алгоритмов хэширования, формирующих 128-битовые хэш-значения.

Однонаправленные хэш-функции на основе симметричных блочных алгоритмов

Однонаправленную хэш-функцию можно построить, используя симметричный блочный алгоритм. Наиболее очевидный подход состоит в том, чтобы шифровать сообщение M посредством блочного алгоритма в режиме CBC или CFB с помощью фиксированного ключа и некоторого вектора инициализации IV. Последний блок шифртекста можно рассматривать в качестве хэш-значения сообщения M. При таком подходе не всегда возможно построить безопасную однонаправленную хэш-функцию, но всегда можно получить код аутентификации сообщения MAC (MessageAuthenticationCode).

Более безопасный вариант хэш-функции можно получить, используя блок сообщения в качестве ключа, предыдущее хэш-значение – в качестве входа, а текущее хэш-значение – в качестве выхода. Реальные хэш-функции проектируются еще более сложными. Длина блока обычно определяется длиной ключа, а длина хэш-значения совпадает с длиной блока.

Поскольку большинство блочных алгоритмов являются 64-битовыми, некоторые схемы хэширования проектируют так, чтобы хэш-значение имело длину, равную двойной длине блока.

Если принять, что получаемая хэш-функция корректна, безопасность схемы хэширования базируется на безопасности лежащего в ее основе блочного алгоритма. Схема хэширования, у которой длина хэш-значения равна длине блока, показана на рисунке 3. Ее работа описывается выражениями:

Практическая работа № 1. Электронная цифровая подпись

2. Основы теории

Таблица 1 – Схемы безопасного хэширования, у которых длина хэш-значения равна длине блока

Номер схемы	Функция хеширования
1	$H_i = E_{H_{i-1}}(M_i) \oplus M_i$
2	$H_i = E_{H_{i-1}}(M_i \oplus H_{i-1}) \oplus M_i \oplus H_{i-1}$
3	$H_i = E_{H_{i-1}}(M_i) \oplus H_{i-1} \oplus M_i$
4	$H_i = E_{H_{i-1}}(M_i \oplus H_{i-1}) \oplus M_i$
5	$H_i = E_{M_i}(H_{i-1}) \oplus H_{i-1}$
6	$H_i = E_{M_i}(M_i \oplus H_{i-1}) M_i \oplus H_{i-1}$
7	$H_i = E_{M_i}(H_{i-1}) M_i \oplus H_{i-1}$
8	$H_i = E_{M_i}(M_i \oplus H_{i-1}) \oplus H_{i-1}$
9	$H_i = E_{M_i \oplus H_{i-1}}(M_i) \oplus M_i$
10	$H_i = E_{M_i \oplus H_{i-1}}(H_{i-1}) \oplus H_{i-1}$
11	$H_i = E_{M_i \oplus H_{i-1}}(M_i) \oplus H_{i-1}$
12	$H_i = E_{M_i \oplus H_{i-1}}(H_{i-1}) \oplus M_i$

Сообщение M разбивается на блоки M_i принятой длины, которые обрабатываются поочередно.

Три различные переменные A, B и C могут принимать одно из четырех возможных значений, поэтому в принципе можно получить 64 варианта общей схемы этого типа. Из них 52 варианта являются либо тривиально слабыми, либо небезопасными. Остальные 12 безопасных схем хэширования перечислены в таблице 1.

Первые четыре схемы хэширования, являющиеся безопасными при всех атаках, приведены на рисунке 4.

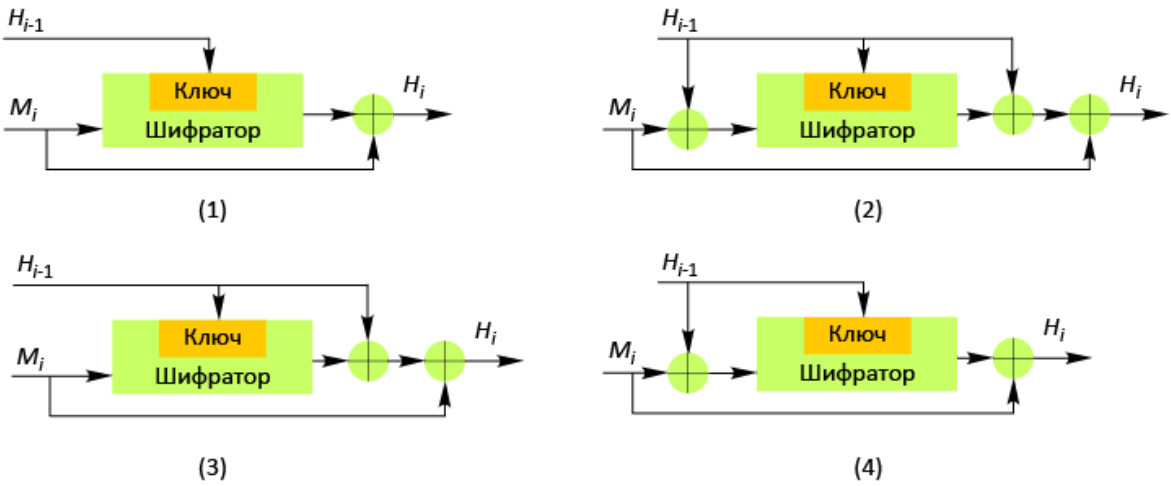

Рисунок 4 – Четыре схемы безопасного хэширования

Практическая работа № 1. Электронная цифровая подпись

3. Задание

1. Изучить материал по вопросу "Электронная цифровая подпись".
2. Написать реферат на тему, соответствующую Вашему индивидуальному варианту.
3. Номер варианта соответствует последней цифре номера Вашей зачетной книжки.
4. В реферате обязательно должен быть список использованной литературы и ссылки на нее в материале реферата.
5. Сдать работу преподавателю.

Практическая работа № 1. Электронная цифровая подпись

3. Задание

Практическая работа № 1. Электронная цифровая подпись

4. Индивидуальные варианты

- Вариант 0. Тема "Сравнительный анализ алгоритмов RSA и DSA"
- Вариант 1. Тема "Алгоритм цифровой подписи RSA"
- Вариант 2. Тема "Алгоритм цифровой подписи DSA"
- Вариант 3. Тема "Алгоритм цифровой подписи ECDSA"
- Вариант 4. Тема "ГОСТ Р 34.10-94. Обзор,назначение, область применения, особенности"
- Вариант 5. Тема "ГОСТ Р 34.10-2001 Обзор,назначение, область применения, особенности"
- Вариант 6. Тема "Сравнительный анализ алгоритмов DSA и ECDSA"
- Вариант 7. Тема "Модели атак на цифровую подпись и их возможные результаты"
- Вариант 8. Тема "Управление ключами в цифровой подписи"
- Вариант 9. Тема "Области использования цифровой электронной подписи"

Практическая работа № 1. Электронная цифровая подпись

4. Индивидуальные варианты