УДК 336.717:004.78

Кузнецов А.А., д.т.н, с.н.с. (Харьковский университет ВС)
Евсеев С.П., к.т.н. (Харьковский национальный экономический университет)
Ткачов А.М., к.т.н. (Харьковский университет ВС)
Король О.Г. (Харьковский национальный экономический университет)

АНАЛИЗ МЕХАНИЗМОВ ОБЕСПЕЧЕНИЯ АУТЕНТИЧНОСТИ И ЦЕЛОСТНОСТИ ДАННЫХ В БАНКОВСКИХ ПЛАТЕЖНЫХ СИСТЕМАХ

Введение. Электронные документы повсеместно используются в корпоративной среде и в государственном управлении, вытесняя бумажные аналоги. Традиционные бумажные технологии делопроизводства и архивирования постепенно заменяются электронным документооборотом. Однако сегодня, в условиях глобализации, роста объема информации, активизации информационного обмена, электронные документы подвергаются серьезным угрозам безопасности. Так, развитие технологий Web, XML, мобильной и беспроводной связи, стандартизация форматов данных и протоколов их обмена делают информационную среду организаций всё более открытой и беззащитной перед криминализирующимся Интернетом. В таких условиях всё большее значение приобретают технологии защиты систем электронных документов как от внешних, так и от внутренних угроз [1-4].

В настоящее время большинство банков для авторизации электронных документов и банковских транзакций при межбанковских и внутрибанковских расчетах, а также при работе с клиентами используют цифровые подписи и функции хеширования [3 – 8].

Целью статьи является анализ механизмов обеспечения аутентичности и целостности информации в банковских платежных системах (БПС), исследование алгоритмов цифровой подписи и хеширования, оценка их основных вероятностно-временных характеристик.

1. Классификация алгоритмов формирования/проверки ЭЦП.

При рассмотрении стандартов ISO 7498-2 и ISO/IEC 10181 определено, что одним из самых надежных способов решения задач,

связанных с аутентификацией (процедура установления достоверности утверждения о том, что объект (или субъект) обладают заявленными свойствами) данных и источников сообщений, являются процедуры цифровой подписи, построенные на основе ассиметричных криптографических алгоритмов [8]. ЭЦП представляет собой строку данных, которая зависит от некоторого секретного параметра (ключа), известного только подписывающему лицу, и от содержания подписываемого сообщения, представленного в цифровом виде [3 – 8].

Таким образом, цифровая подпись сообщения – это блок данных небольшого размера, полученный в результате криптографического преобразования сообщения произвольной длины с использованием личного ключа отправителя [8], который связывает сообщение с некоторым порождающим или подписывающим его объектом.

Основными стандартами ЭЦП являются [2, 4, 5]: – международный стандарт ISO/IEC 9796, который определяет ЭЦП с восстановлением сообщения (digital signature with message recovery);

- международный стандарт ISO/IEC 14888, который определяет ЭЦП с добавлением (digital signature with appendix);
- российский стандарт цифровой подписи на эллиптической кривой ГОСТ Р 34.10-2001;
- американский национальный стандарт цифровой подписи (FIPS 186);
- американский финансовый стандарт цифровой подписи с добавлением на эллиптической кривой (ANSI X9.62);
- стандарт на ЭЦП PKCS #1, который определяет ЭЦП на основе алгоритма RSA;
- стандарт цифровой подписи с добавлением и восстановлением сообщения IEEE 1363;
- стандарт цифровой подписи с добавлением на эллиптической кривой IEEE P 1363;
- международный стандарт ISO/IEC CD 15946-2 стандартизирует ЭЦП на эллиптической кривой с добавлением;
- государственный стандарт Украины ДСТУ-4154 – 2002.

На основе существующих стандартов ЭЦП в [2] предложена классификация ЭЦП.

По способу построения схемы ЭЦП делятся на два класса:

- схема ЭЦП с восстановлением сообщения;
- схема ЭЦП с добавлением.

В схемах ЭЦП с восстановлением сообщения всё или часть подписанного сообщения может быть восстановлено непосредственно из ЭЦП. Таким образом, на вход алгоритма верификации поступает лишь цифровая подпись.

В схеме ЭЦП с добавлением цифровая подпись присоединяется к сообщению и в таком виде отправляется адресату. Для верификации такой ЭЦП необходимо иметь и подпись, и соответствующее сообщение.

По способу формирования ЭЦП делятся на два класса:

- схема ЭЦП Off-line;
- схема ЭЦП On-line.

В схемах ЭЦП off-line цифровая подпись формируется без участи сторонних лиц или, по крайней мере, не нуждается в канале связи. В схемах ЭЦП on-line подписываемому необходим канал связи с другим лицом для выработки цифровой подписи сообщения.

По времени действия ЭЦП подразделяются:

- схемы ЭЦП без ограничения;
- схемы ЭЦП с ограничением.

В схемах ЭЦП без ограничения подписываемый может сформировать цифровую подпись для ранее неподписанного сообщения в любой момент, либо срок действия цифровой подписи неограничен.

По количеству участников ЭЦП подразделяется:

- одиночная схема ЭЦП;
- групповая схема ЭЦП.

В процессе выполнения алгоритма формирования цифровой подписи в одиночных схемах ЭЦП достаточно одного участника, в групповых схемах их два или больше.

По способу проверки ЭЦП делятся на два класса:

- интерактивные схемы ЭЦП, требующие протокольного взаимодействия;
- не интерактивные схемы ЭЦП, не требующие протокольного взаимодействия.

Существующие алгоритмы ЭЦП можно разделить также по типу используемых однонаправленных функций с секретом. Таких задач большое количество. В данной классификации упоминаются цифровые подписи, которые используются в каком-либо стандарте. Поэтому ЭЦП по типу используемых однонаправленных функций с секретом подразделяются:

- схемы ЭЦП, основанные на стойкости факторизации большого числа;

- схемы ЭЦП, основанные на стойкости дискретного логарифма;
- схемы ЭЦП, основанные на стойкости дискретного логарифма в группе точек ЭК.

Каждая из этих схем может быть детерминированной или рандомизированной. Применение детерминированных схем характеризуется тем, что цифровая подпись одной и той же входной строки данных приводит к формированию одинаковых цифровых подписей. В рандомизированной схеме при генерации подписи используется некоторый случайный параметр, что приводит к формированию различных подписей даже для одинаковых входных строк. В рандомизированных схемах необходимо обеспечить непредсказуемость случайных чисел.

В свою очередь детерминированные схемы делятся на схемы ЭЦП одноразового применения и схемы ЭЦП многократного применения.

Используя данную классификацию можно дать полное описание любой ЭЦП. Например [9] – есть ЭЦП с добавлением, off-line, без ограничения формирования и проверки ЭЦП, стойкость основана на сложности дискретного логарифма в группе точек ЭК. В процессе формирования участвует только одно лицо, а при проверки ЭЦП не требуется наличие канала связи, она является также рандомизированной схемой [2, 4, 5].

Рассмотрим основные стандарты ЭЦП, применяемых в комплексных системах защиты банковской информации, проведем сопоставление основных характеристик ЭЦП (длину ключей, длину цифровой подписи, сложность (время) вычисления и сложность (время) проверки подлинности цифровой подписи) с целью выявления их преимуществ и недостатков, при условии, что уровень стойкости подписи по отношению к любым методам фальсификации не ниже, чем 10^{21} (или 30 лет непрерывной работы сети из 1000 суперкомпьютеров).

2. Анализ алгоритмов цифровой подписи в современных банковских системах.

В качестве "базовой" длины ключей и длины самой цифровой подписи мы будем рассматривать длину в 64 байта.

Алгоритм RSA [2 – 5] основывается на NP-полной задаче нахождения дискретного логарифма [4, 10] (разложения целого параметра n в виде произведения двух различных простых чисел примерно равных по порядку величины, т.е. $n = p \times q$ на эти простые множители). По современным оценкам сложность задачи разложения на простые множители при целых числах n из 64 байт составляет порядка 10^{17} - 10^{18}

операций, т. е. находится где-то на грани досягаемости для серьезного "взломщика". Поэтому обычно в системах цифровой подписи на основе алгоритма RSA применяют более длинные целые числа n (обычно от 75 до 128 байт). Это соответственно приводит к увеличению длины самой цифровой подписи относительно 64-байтного варианта примерно на 20% - 100% (в данном случае ее длина совпадает с длиной записи числа n), а также от 70% до 800% увеличивает время вычислений при подписывании и проверке.

Кроме того, при генерации и вычислении ключей в системе RSA необходимо проверять большое количество довольно сложных дополнительных условий на простые числа p и q, а невыполнение любого из них может сделать возможным фальсификацию подписи со стороны того, кто обнаружит невыполнение хотя бы одного из этих условий (при подписывании важных документов допускать, даже теоретически, такую возможность нежелательно) [2, 4, 5].

Алгоритм EGSA [2, 4, 5]. Существенным шагом вперед в разработке современных алгоритмов цифровой подписи был новый алгоритм Эль Гамаля. В этом алгоритме целое число n полагается равным специально выбранному большому простому числу p, по модулю которого и производятся все вычисления. Такой выбор позволяет повысить стойкость подписи при ключах из 64 байт примерно в 1000 раз, т.е. при такой длине ключей обеспечивается необходимый нам уровень стойкости порядка 10^{21}. Правда, при этом длина самой цифровой подписи увеличивается в два раза и составляет 128 байт. Главная "заслуга" алгоритма Эль Гамаля состояла в том, что в дальнейшем он послужил основой для принятия нескольких стандартов цифровой подписи, в том числе национального стандарта США DSS и государственного стандарта РФ ГОСТ Р-34.10-2001.

Алгоритм DSA [2, 4, 5]. Алгоритм DSA, ставший в дальнейшем основой национального стандарта США на цифровую подпись имеет по сравнению с алгоритмом RSA целый ряд преимуществ:

- при заданном уровне стойкости цифровой подписи целые числа, с которыми приходится проводить вычисления, имеют запись как минимум на 20% короче, что соответственно уменьшает сложность вычислений не менее, чем на 70% и позволяет заметно сократить объем используемой памяти;

- при выборе параметров достаточно проверить всего три достаточно легко проверяемых условия;

- процедура подписывания по этому методу не позволяет вычислять (как это возможно в RSA) цифровые подписи под новыми сообщениями без знания секретного ключа.

По сравнению с оригинальным алгоритмом Эль Гамаля метод DSA имеет одно важное преимущество, - при заданном в стандарте уровне стойкости, числа, участвующие в вычислении подписи, имеют длину по 20 байт каждое, сокращая общую длину подписи до 40 байт.

Поскольку большинство операций при вычислении подписи и ее проверке также производится по модулю из 20 байт, сокращается время вычисления подписи и объем используемой памяти.

В алгоритме Эль Гамаля длина подписи при таком уровне стойкости была бы равна 128 байт.

Алгоритм НОТАРИУС-1 – аналог алгоритма Эль Гамаля. Основное отличие алгоритма состоит в том, что вместо обычной операции умножения целых чисел по модулю большого простого p используется умножение по модулю простого числа, эта операция гораздо эффективней вычисляется на распространенных процессорах. Процедуры подписывания электронных документов и проверки цифровых подписей по алгоритму НОТАРИУС-1 выглядят аналогично соответствующим процедурам алгоритма Эль Гамаля, обеспечивают тот же уровень стойкости подписи, но выполняются быстрее [11].

Алгоритм НОТАРИУС-S позволил при сохранении стойкости подписи сократить ее длину еще на 32.5%. Для базового варианта с ключами из 64 байт длина подписи сократилась относительно DSA и НОТАРИСА-D с 40 байт до 27 байт. Соответственно уменьшилось время вычисления и проверки подписи. Стойкость осталась на том же уровне - 10^{21}.

Алгоритм ЭЦП ГОСТ34.10 – 2001. Алгоритм вычисления и проверки подписи в ГОСТ34.10 устроен аналогично алгоритму DSA, но предварительная обработка электронных документов перед подписыванием (так называемое хэширование) выполняются по другому, существенно более медленному способу [1, 2].

Алгоритм ЭЦП ДСТУ-4145. Алгоритм вычисления и проверки ЭЦП основанный на свойствах групп точек эллиптической кривой над полями GF (2^m). Стойкость алгоритма основана на NP-полной задаче нахождения дискретного логарифма эллиптической кривой [8] (нахождения значения k по базовой точке P и расположенной на кривой точке kP). При этом алгоритмы на основе эллиптической кривой используют ключи малых

размеров, что снижает требования к вычислительным мощностям по сравнению с требованиями к алгоритмам на основе RSA.

В таблице 1 представлены сравнительные характеристики алгоритмов RSA и ECDSA (нечетный случай) при создании и проверке цифровых подписей. Оба алгоритма тестировали на параллельных процессорах Motorola 56303 DSP (66 МГц) [1]. При этом функция проверки подписи RSA использует e = 65 537.

Таблица 1 - Сравнительные характеристики алгоритмов RSA и ECDSA (нечетный случай) при создании и проверке цифровой подписи

Алгоритм (длина ключа, бит)	Время выполнения, мс	
	Создание подписи	Проверка подписи
RSA (1024)	25	< 2
ECDSA (160)	32	33
RSA (2048)	120	5
ECDSA (216)	68	70

Как видно из таблицы 1, при увеличении размеров ключа создание подписей с помощью ECDSA производится значительно быстрее, чем в системах RSA. Это различие еще в большей степени проявляется для однопроцессорных систем. С другой стороны, проверка подписи с помощью ECDSA, делается намного медленнее, чем эта же процедура в системах RSA и опять же различие усиливается для систем с одним процессором [1].

Обработка ECDSA может несколько ускориться в "четном" случае. Мощность процессора, затраченная на проверку подписи ECDSA, может замедлить выполнение других приложений в системе. В некоторых случаях системы RSA (даже использующие большие ключи) возможно, будут более приемлемы, чем криптосистемы на основе эллиптической кривой. Тем не менее, криптосистемы на основе эллиптической кривой получают все большее распространение скорее как альтернатива, а не замена систем RSA, поскольку системы ECDLP имеют некоторые преимущества, особенно при использовании в устройствах с маломощными процессорами и/или маленькой памятью [1].

В таблице 2 приведены результаты сравнительного анализа наиболее распространенных алгоритмов ЭЦП.

Таблица 2 - Результаты сравнительного анализа наиболее распространенных алгоритмов ЭЦП

Алгоритм	"Нотариус-S"			NIST DSA			ГОСТ 34.10		
P Длина ключа (байт)	L Длина подписи (байт)	Q Сложность подделки подписи без ключа	T Время подделки подписи без ключа	L Длина подписи (байт)	Q Сложность подделки подписи без ключа	T Время подделки подписи без ключа	L Длина подписи (байт)	Q Сложность подделки подписи без ключа	T Время подделки подписи без ключа
16	15	$2*10^{14}$	10 дней	-	-	-	-	-	-
18	16	$5*10^{14}$	1 месяц	-	-	-	-	-	-
21	17	$8*10^{15}$	1 год	-	-	-	-	-	-
24	18	$7*10^{16}$	10 лет	-	-	-	-	-	-
36	21	$5*10^{19}$	2 года	-	-	-	-	-	-
48	24	10^{22}	400 лет	-	-	-	-	-	-
64	27	10^{24}	2недели	40	10^{24}	2недели	64	10^{24}	2недели
80	30	$7*10^{26}$	25 лет	40	10^{24}	2недели	-	-	-
104	33	$5*10^{29}$	18000 лет	40	10^{24}	2недели	-	-	-
128	36	$2*10^{31}$	800000 лет	40	10^{24}	2недели	64	$2*10^{31}$	800000 лет
160	39	$9*10^{34}$	3.9 млн. лет	-	-	-	-	-	-
192	42	$3*10^{37}$	10^{7}лет	-	-	-	-	-	-
224	45	$5*10^{39}$	Более 100 млрд. лет						
256	48	$6*10^{41}$							
304	51	$4*10^{44}$							
352	54	$1.5*10^{47}$							
400	57	$3*10^{49}$							
448	60	$4*10^{51}$							
512	63	$2*10^{54}$							

Проведенный анализ данных таблицы 1 позволяет сделать вывод, что основным методом защиты ЭЦП является увеличение параметров модулей преобразования порядка 512 и более битов. Но при этом до такой же длины увеличиваются и длины ключей. Как следствие увеличивается вычислительная сложность криптографических преобразований и уменьшается скорость. В тоже время все преобразования в банковских системах необходимо осуществлять в реальном масштабе времени обеспечивая требуемые показатели конфиденциальности и целостности данных.

3. Процедуры построения электронных цифровых подписей.

Система ЭЦП включает две процедуры: 1) процедуру генерации подписи; 2) процедуру верификации подписи. В процедуре генерации

подписи используется секретный ключ отправителя сообщения, в процедуре верификации подписи – открытый ключ отправителя. Секретный ключ хранится абонентом в тайне и используется им для формирования ЭЦП. Открытый ключ известен всем пользователям сети [12, 13].

При формировании ЭЦП отправитель вычисляет хэш-функцию h(M) подписываемого текста M, предназначенную для сжатия и перемешивания подписываемого документа M до нескольких десятков или сотен бит (фиксированной длины). Вычисленное значение хэш-функции h(M) представляет собой один короткий блок информации m (образ), характеризующий весь текст M в целом. Затем число m шифруется на личном ключе отправителя. Получаемая при этом пара чисел (необязательно) представляет собой ЭЦП для данного текста M (рисунок 1).

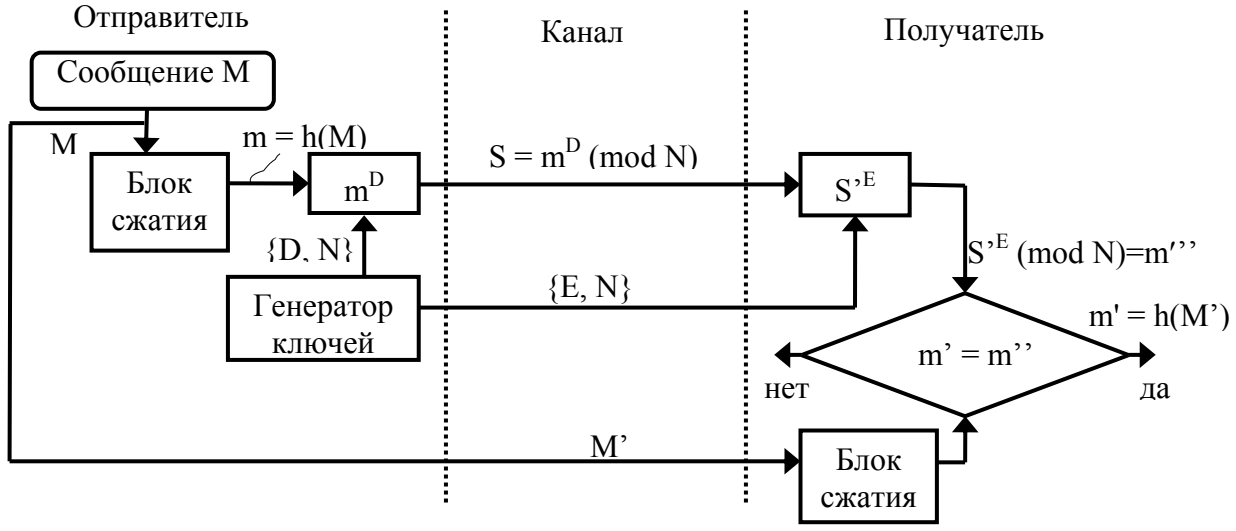

Рисунок 1 - Схема формирования и верификации ЭЦП методом RSA

При верификации ЭЦП получатель сообщения снова вычисляет хэш-функцию m' = h(M') принятого по каналу исходного текста M (возможно измененного), после чего при помощи открытого ключа E отправителя проверяет, соответствует ли полученная подпись вычисленному значению хэш-функции m' = m'' [5, 12].

Достаточно эффективным механизмом для обеспечения аутентичности сообщений является однонаправленные хэш-функции. Часть из них строится на основе симметричного блочного алгоритма шифрования в режиме CBC или CPB, с помощью фиксированного ключа и

некоторого вектора инициализации IV. Последний блок шифртекста и есть хэш-значения сообщения M. При таком подходе не всегда возможно построить безопасную однонаправленную хэш-функцию, но всегда можно получить код аутентификации сообщения MAC (Message Authentication Code). Основным преимуществом этого механизма в сравнении с ЭЦП является более простой алгоритм генерации и верификации, что позволяет обеспечивать высокое быстродействие алгоритмов аутентификации сообщений в беспроводных сетях передачи данных.

Для таких механизмов длина блока определяется длиной ключа, а длина хэш-значения совпадает с длиной блока. Четыре из наиболее распространенных схем хеширования, являющиеся безопасными при всех атаках, приведены на рисунке 2.

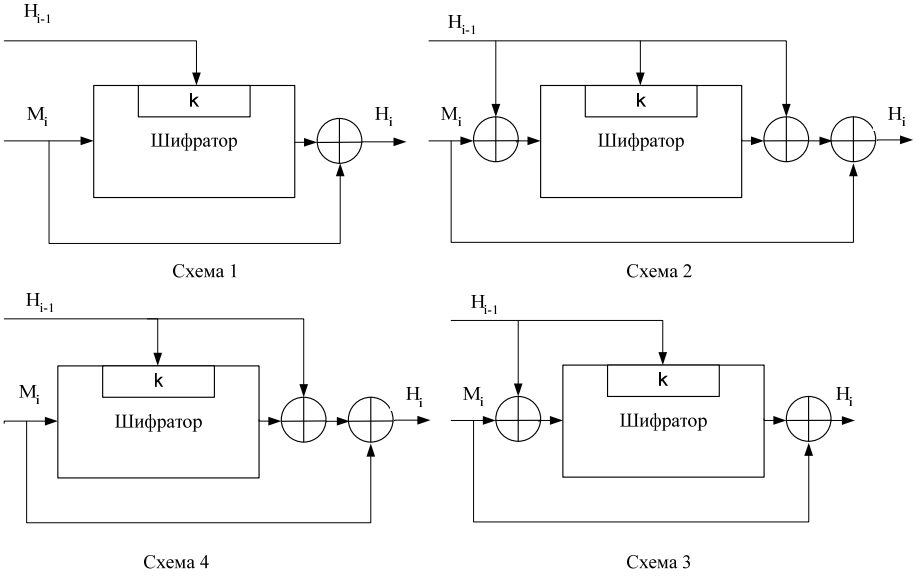

Рисунок 2 - Наиболее распространенные варианты схем хеширования

Из представленных схем (рисунок 2) при фиксированном шифре более стойкой является схема 2, однако она является более сложной в реализации. Последнее обусловлено результатом хеширования на предыдущем шаге h_{i-1}, который подается в качестве входа цикловой функции, в качестве ключа на i-ом шаге k_{i-1} и складывается с результатом h_i. Теоретическое доказательство вероятности коллизий в таком случае является тривиальным [6].

Для примера рассмотрим одну из распространенных функций хеширования ГОСТ 34.311-95. ГОСТ 34.311-95 используется в современных программных средствах "Грифон-Б" и "Грифон-Л" (ООО

СНПФ "АРГУС"). последовательного хэширования с фиксированным размером входа (функция сжатия с коэффициентом). На рисунке 3 представлена обобщенная структурная схема функции хеширования по ГОСТ 34.311 – 95.

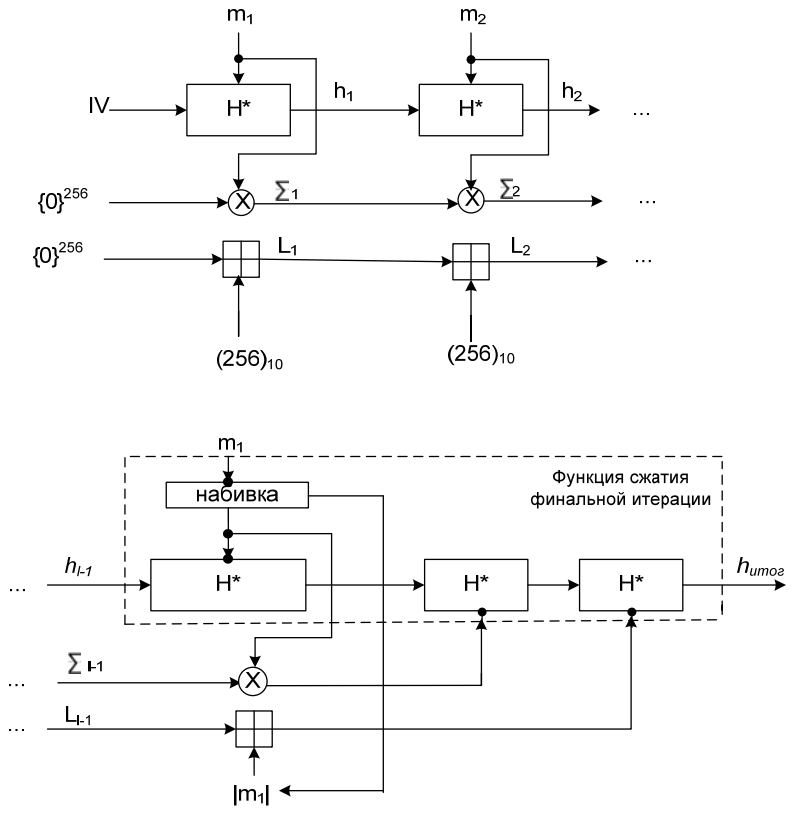

Рисунок 3 - Обобщенная структурная схема функции хеширования ГОСТ 34.311 – 95

При этом хэширование сообщения m производится в последовательности:

$$h \leftarrow IV, \quad h_i \leftarrow H\left(m_i, h_{i-1}\right) \quad \text{для } i = 1, 2, \ldots l, \quad h_{\text{итог}} \leftarrow h_l,$$

где H_i – функция сжатия, а h_i – переменная сцепления.

При необходимости последний блок заполняется до длины кратной n. В отличие от стандартных предпосылок в ГОСТ 34.311 – 95 процедура хэширования ожидает конца сообщения, а после делается "набивка".

Анализ [2] хэш-функции по ГОСТ 34.311 – 95 позволяет сделать следующие выводы:

- булева функция S-box-ов линейна и ее применение неоправданно – это приводит к снижению скорости обработки данных из-за большого числа повторений перемешивающего преобразования для достижения "заданного" уровня безопасности;

- шифрующее преобразование, при определенных допущениях, невозможно атаковать по частям, а, следовательно, функцию сжатия можно считать стойкой к столкновениям;

- алгоритм хэширования является методом последовательного хэширования с MD – усилением (коэффициент сжатия 2);

- стойкость хэш-функции в известной мере зависит от выбора блоков замен в шифрующем преобразовании, к тому же они один из параметров инициализации алгоритма хэширования;

- IV в стандарте не фиксирован, а это подразумевает, что необходимо выработать правила его использования, к тому же имеется большой класс атак на псевдостолкновения при нефиксированном IV;

- скорость обработки данных хэш-функцией значительно меньше, чем у аналогичных по внешним параметрам HAVAL, SHA-256, а тем более остального MD-семейства, из-за попыток ликвидировать очевидные оплошности конструирования усложнением функции сжатия;

- приблизительная скорость реализации 4/5 от скорости реализации лежащего в основе алгоритма шифрования [14];

- с учетом парадокса дней рождения вычислительная сложность нахождения коллизии составляет $2^{256/2}$ операций хэширования.

Это позволяет заявить о более высокой криптографической стойкости алгоритма в сравнении HAVAL, SHA-256 и потенциально высокой стойкости к коллизиям. Однако, на сегодняшний день найдена коллизия для функций MD-5, SHA-1 подобных схеме ГОСТ 34.311 – 95. Somitra Kumar Sanadhya и Palash Sarkary [1] получили новый метод определения коллизии на 22 шаге хеш-функции SHA-2 с вероятностью 2^{-5} и 2^{-9}. Не смотря на то, что SHA-256 имеет 64 раунда, а SHA-512 – 80 раундов задача определения коллизии не является тривиальной.

Выводы. Проведенные исследования показывают, что одним из наиболее эффективных механизмов обеспечения целостности и аутентичности информации в современных банковских системах является ЭЦП. При формировании ЭЦП отправитель вычисляет хэш-функцию подписываемого документа, предназначенную для сжатия и перемешивания. Другими словами, такие показатели эффективности ЭЦП как криптографическая стойкость и вычислительная сложность реализации непосредственно определяются конструктивными особенностями

применяемой функции хеширования. Однако, как показал проведенный анализ, на сегодняшний день Украина не имеет Национального стандарта хеширования информации, для обеспечения целостности и аутентичности данных используются алгоритмы, определенные международными и Российскими стандартами. Таким образом, разработка и исследование перспективных методов и алгоритмов ключевого хеширования является актуальным направлением дальнейших исследований.

Список литературы

1. Соколов А.В., Степанюк О.М. Защита от компьютерного терроризма. – БХВ-Петербург Арлит. 2002. – 496 с.

2. Домарев В.В. Защита информации и безопасность компьютерных систем. – Киев., Издательство "ДиаСофт". 1999. – 480 с.

3.Артеменко Д. А. Механизм обеспечения финансовой безопасности банковской деятельности : Дис. канд. экон. наук: 08.00.10 : Ростов н/Д, 1999. – 190 с.

4. Чмора А. Л. Современная прикладная криптография. –Москва. 2002. – 508 с.

5. В. Столлингс Криптография и защита сетей: принципы и практика, 2-е изд. : пер. с англ. — М.: издательский дом «Вильям», 2001. — 672 с.

6. Євсеєв С.П., Чевардин В.Е., Радковский С.А. Механизмы обеспечения аутентичности банковских данных во внутриплатежных системах комерческого банка. / Збірник наукових статей ХНЕУ. – Харків: ХНЕУ. – 2008. – Вип. 6. – С. 40-44.

7. Кузнецов А.А., Король О.Г., Ткачов А.М. Анализ механизмов обеспечения безопасности банковской информации во внутриплатежных системах коммерческого банка / Матеріали І міжнародної науково-практичної конференції «Безпека та захист інформації в інформаційних і телекомунікаційних системах» 28 – 29 травня 2008 р. Зб. наук. статей «Управління розвитком». ХНЕУ. № 6 – Х.: 2008. – С. 28 – 35.

8. Информационная технология. Криптографическая защита информации. Цифровая подпись, основанная на эллиптических кривых. Формирование и проверка. ДСТУ 4145 – 2002. – Чинний від 01.01.2002. – К.: Держстандарт України, 2002. – 34 с.

9. Ездаков А., Макарова О. Как защитить информацию // Журнал "Сети". Москва. – 08\97.

10. Разборов А. А. Лекция "Основы теории сложности вычислений". – 1998 г.

11. http://www.infocity.kiev.ua

12. Романец Ю.В., Тимофеев П.А., Шаньгин В.Ф. Защита информации в компьютерных системах и сетях / Под ред. В.Ф. Шаньгина.-2-е изд., перераб. и доп.- М.: Радио и связь, 2001. – 376 с.

13. Логинов А.А., Елхимов Н.С. Общие принципы функционирования электронных платежных систем и осуществление мер безопасности при защите от злоупотребления и мошенничества // Конфидент.-1995.-№4.-С.48-54

14. Шефановский Д.Б. ГОСТ 34.11 – 94. Функция хэширования. Краткий анализ. Учебный центр "Инфозащита". 2001. – 9с.