ГОСУДАРСТВЕННЫЙ СТАНДАРТ
РЕСПУБЛИКИ БЕЛАРУСЬ

СТБ 34.101.31-2011

Информационные технологии и безопасность
Защита информации

КРИПТОГРАФИЧЕСКИЕ АЛГОРИТМЫ ШИФРОВАНИЯ И КОНТРОЛЯ ЦЕЛОСТНОСТИ

Інфармацыйныя тэхналогіі і бяспека
Ахова інфармацыі

КРЫПТАГРАФІЧНЫЯ АЛГАРЫТМЫ ШЫФРАВАННЯ І КАНТРОЛЮ ЦЭЛАСНАСЦІ

Госстандарт
Минск

УДК 004.056.55(083.74)(476) МКС 35.240.40 КП 05

Ключевые слова: информационные технологии, защита информации, криптографический алгоритм, шифрование, контроль целостности, имитозащита, хэширование, управление ключами

Предисловие

Цели, основные принципы, положения по государственному регулированию и управлению в области технического нормирования и стандартизации установлены Законом Республики Беларусь «О техническом нормировании и стандартизации».

1 РАЗРАБОТАН учреждением Белорусского государственного университета «Научно-исследовательский институт прикладных проблем математики и информатики»

ВНЕСЕН Оперативно-аналитическим центром при Президенте Республики Беларусь

2 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ постановлением Госстандарта Республики Беларусь от 31 января 2011 г. № 5

3 ВЗАМЕН СТБ П 34.101.31-2007

Издан на русском языке

Содержание

ГОСУДАРСТВЕННЫЙ СТАНДАРТ РЕСПУБЛИКИ БЕЛАРУСЬ

Информационные технологии. Защита информации

КРИПТОГРАФИЧЕСКИЕ АЛГОРИТМЫ ШИФРОВАНИЯ И КОНТРОЛЯ ЦЕЛОСТНОСТИ

Інфармацыйныя тэхналогіі. Ахова інфармацыі

КРЫПТАГРАФІЧНЫЯ АЛГАРЫТМЫ ШЫФРАВАННЯ І КАНТРОЛЮ ЦЭЛАСНАСЦІ

Information technology and security

Data encryption and integrity algorithms

Дата введения 2011-07-01

1 Область применения

Настоящий стандарт определяет семейство криптографических алгоритмов шифрования и контроля целостности, которые используются для защиты информации при ее хранении, передаче и обработке.

Настоящий стандарт применяется при разработке средств криптографической защиты информации.

2 Нормативные ссылки

В настоящем стандарте использованы ссылки на следующий стандарт:

ГОСТ 34.973-91 (ИСО 8824-87) Информационная технология. Взаимосвязь открытых систем. Спецификация абстрактно-синтаксической нотации версии 1 (АСН.1).

Примечание — При пользовании настоящим стандартом целесообразно проверить действие технических нормативных правовых актов в области технического нормирования и стандартизации (далее — ТНПА) по каталогу, составленному по состоянию на 1 января текущего года, и по соответствующим информационным указателям, опубликованным в текущем году. Если ссылочные ТНПА заменены (изменены), то при пользовании настоящим стандартом следует руководствоваться замененными (измененными) ТНПА. Если ссылочные ТНПА отменены без замены, то положение, в котором дана ссылка на них, применяется в части, не затрагивающей эту ссылку.

3 Термины и определения

В настоящем стандарте применяются следующие термины с соответствующими определениями:

3.1 блок: Двоичное слово длины 128.

3.2 заголовок ключа: Блок, содержащий открытые атрибуты ключа.

3.3 зашифрование: Преобразование сообщения, направленное на обеспечение его конфиденциальности, которое определяется с использованием ключа.

3.4 имитовставка: Двоичное слово, которое определяется по сообщению с использованием ключа и служит для контроля целостности и подлинности сообщения.

3.5 имитозащита: Контроль целостности сообщений, который реализуется путем выработки и проверки имитовставок.

3.6 ключ (секретный): Параметр, который управляет операциями шифрования и имитозащиты и который известен только определенным сторонам.

3.7 октет: Двоичное слово длины 8.

3.8 расширение ключа: Дополнение ключа новыми символами до получения ключа определенной длины.

3.9 расшифрование: Преобразование, обратное зашифрованию.

3.10 синхропосылка: Открытые входные данные криптографического алгоритма, которые обеспечивают уникальность результатов криптографического преобразования на фиксированном ключе.

3.11 снятие защиты: Проверка имитовставок и расшифрование.

3.12 сообщение: Двоичное слово конечной длины.

3.13 преобразование ключа: Построение по исходному ключу набора новых ключей с различными заголовками.

3.14 установка защиты: Зашифрование и вычисление имитовставок.

3.15 шифрование: Зашифрование или расшифрование.

3.16 хэш-значение: Двоичное слово фиксированной длины, которое определяется по сообщению без использования ключа и служит для контроля целостности сообщения и для представления сообщения в сжатой форме.

3.17 хэширование: Выработка хэш-значений.

4 Обозначения

4.1 Список обозначений

$\{0,1\}^n$	множество всех слов длины n в алфавите $\{0,1\}$;
$\{0,1\}^*$	множество всех слов конечной длины в алфавите $\{0,1\}$ (включая пустое слово длины 0);
$\|u\|$	длина слова $u \in \{0,1\}^*$;
$\{0,1\}^{n*}$	множество всех слов из $\{0,1\}^*$, длина которых кратна n;
α^n	слово длины n из одинаковых символов $\alpha \in \{0,1\}$;
$L_m(u)$	слово из первых m символов слова u, $m \leqslant \|u\|$;
$u \parallel v$	конкатенация $u_1u_2\dots u_nv_1v_2\dots v_m$ слов $u = u_1u_2\dots u_n$ и $v = v_1v_2\dots v_m$;
$\mathtt{01234\dots}_{16}$	представление $u \in \{0,1\}^{4*}$ шестнадцатеричным словом, при котором последовательным четырем символам u соответствует один шестнадцатеричный символ (например, $10100010 = \mathtt{A2}_{16}$);
$U \bmod m$	для целого U и натурального m остаток от деления U на m;
$u \oplus v$	для $u = u_1u_2\dots u_n \in \{0,1\}^n$ и $v = v_1v_2\dots v_n \in \{0,1\}^n$ слово $w = w_1w_2\dots w_n \in \{0,1\}^n$ из символов $w_i = (u_i + v_i) \bmod 2$;
$\bar{u}$	а) для $u = u_1u_2\dots u_8 \in \{0,1\}^8$ число $2^7u_1 + 2^6u_2 + \dots + u_8$ и б) для $u = u_1 \parallel u_2 \parallel \dots \parallel u_n$, $u_i \in \{0,1\}^8$, число $\bar{u}_1 + 2^8\bar{u}_2 + \dots + 2^{8(n-1)}\bar{u}_n$;

$\langle U \rangle_{8n}$	для целого U слово $u \in \{0,1\}^{8n}$ такое, что $\bar{u} = U \bmod 2^{8n}$;
$u \boxplus v$	для $u, v \in \{0,1\}^{8n}$ слово $\langle \bar{u} + \bar{v} \rangle_{8n}$;
$u \boxminus v$	для $u, v \in \{0,1\}^{8n}$ слово $w \in \{0,1\}^{8n}$ такое, что $u = v \boxplus w$;
$\lfloor z \rfloor$	для вещественного z максимальное целое, не превосходящее z;
$\lceil z \rceil$	для вещественного z минимальное целое, не меньшее z;
$\mathtt{ShLo}(u)$	для $u \in \{0,1\}^{8n}$ слово $\langle \lfloor \bar{u}/2 \rfloor \rangle_{8n}$;
$\mathtt{ShHi}(u)$	для $u \in \{0,1\}^{8n}$ слово $\langle 2\bar{u} \rangle_{8n}$;
$\varphi^r(u)$	для слова u и преобразования φ результат r-кратного действия φ на u (например, $\mathtt{ShLo}^r(u)$ — результат r-кратного действия $\mathtt{ShLo}$);
$\mathtt{RotHi}(u)$	для $u \in \{0,1\}^{8n}$ слово $\mathtt{ShHi}(u) \oplus \mathtt{ShLo}^{8n-1}(u)$;
$\mathbb{F}_2$	поле из двух элементов 0 и 1;
$\mathbb{F}_2[x]$	кольцо многочленов над полем $\mathbb{F}_2$;
$u(x)$	а) для $u = u_1 u_2 \dots u_8 \in \{0,1\}^8$ многочлен $u_1 x^7 + u_2 x^6 + \dots + u_8$ и б) для $u = u_1 \parallel u_2 \parallel \dots \parallel u_n$, $u_i \in \{0,1\}^8$, многочлен $u_1(x) + x^8 u_2(x) + \dots + x^{8(n-1)} u_n(x)$;
$u(x) \bmod f(x)$	для $u(x) \in \mathbb{F}_2[x]$ и ненулевого $f(x) \in \mathbb{F}_2[x]$ остаток от деления $u(x)$ на $f(x)$;
$u * v$	для $u, v \in \{0,1\}^{128}$ слово $w \in \{0,1\}^{128}$ такое, что $w(x) = u(x)v(x) \bmod x^{128} + x^7 + x^2 + x + 1$;
$a \leftarrow u$	присвоение переменной a значения u;
$a \leftrightarrow b$	перестановка значений переменных a и b;
$F_\theta(X)$	результат зашифрования блока $X \in \{0,1\}^{128}$ на ключе $\theta \in \{0,1\}^{256}$ по алгоритму из 6.1.3;
$F_\theta^{-1}(X)$	результат расшифрования блока $X \in \{0,1\}^{128}$ на ключе $\theta \in \{0,1\}^{256}$ по алгоритму из 6.1.4.

4.2 Пояснения к обозначениям

4.2.1 Слова

Входными и выходными данными алгоритмов настоящего стандарта являются двоичные слова. Двоичные слова представляют собой последовательности символов из алфавита $\{0,1\}$. Символы нумеруются слева направо от единицы. В настоящем подразделе в качестве примера рассматривается слово

$$w = 10110001100101001011101011001000.$$

В этом слове первый символ — 1, второй — 0, ..., последний — 0.

Слова разбиваются на тетрады из четверок последовательных двоичных символов. Тетрады кодируются шестнадцатеричными символами по следующим правилам (см. таблицу 1):

Таблица 1

тетрада	символ	тетрада	символ	тетрада	символ	тетрада	символ
0000	$\mathtt{0}_{16}$	0001	$\mathtt{1}_{16}$	0010	$\mathtt{2}_{16}$	0011	$\mathtt{3}_{16}$
0100	$\mathtt{4}_{16}$	0101	$\mathtt{5}_{16}$	0110	$\mathtt{6}_{16}$	0111	$\mathtt{7}_{16}$
1000	$\mathtt{8}_{16}$	1001	$\mathtt{9}_{16}$	1010	$\mathtt{A}_{16}$	1011	$\mathtt{B}_{16}$
1100	$\mathtt{C}_{16}$	1101	$\mathtt{D}_{16}$	1110	$\mathtt{E}_{16}$	1111	$\mathtt{F}_{16}$

Например, слово w кодируется следующим образом:

$$\mathtt{B194BAC8}_{16}.$$

Пары последовательных тетрад образуют октеты. Последовательные октеты слова w имеют вид:

$$10110001 = \mathtt{B1}_{16},\ 10010100 = \mathtt{94}_{16},\ 10111010 = \mathtt{BA}_{16},\ 11001000 = \mathtt{C8}_{16}.$$

4.2.2 Слова как числа

Октету $u = u_1 u_2 \dots u_8$ ставится в соответствие байт — число $\bar{u} = 2^7 u_1 + 2^6 u_2 + \dots + u_8$. Например, октетам w соответствуют байты

$$177 = 2^7 + 2^5 + 2^4 + 1,\ 148 = 2^7 + 2^4 + 2^2,\ 186 = 2^7 + 2^5 + 2^4 + 2^3 + 2^1,\ 200 = 2^7 + 2^6 + 2^3.$$

Число ставится в соответствие не только октетам, но и любому другому двоичному слову, длина которого кратна 8. При этом используется распространенное для многих современных процессоров соглашение «от младших к старшим» (little-endian): считается, что первый байт является младшим, последний — старшим. Например, слову w соответствует число

$$\bar{w} = 177 + 2^8 \cdot 148 + 2^{16} \cdot 186 + 2^{24} \cdot 200 = 3367670961.$$

При отождествлении слов с числами удобно представить себе гипотетический регистр, разрядность которого совпадает с длиной слова. В самый правый октет регистра загружается первый октет слова, во второй справа октет регистра — второй октет слова и так далее, пока, наконец, в самый левый октет регистра не загружается последний октет слова. Например, для w содержимое регистра имеет вид:

$$\mathtt{C8BA94B1}_{16} = 11001000101110101001010010110001.$$

При таком представлении операции `ShLo`, `ShHi`, `RotHi` состоят в сдвигах содержимого регистра: `ShLo` — вправо (в сторону младших разрядов), `ShHi` — влево (в сторону старших разрядов) и `RotHi` — циклически влево, причем при сдвигах `ShLo` и `ShHi` в освободившиеся разряды регистров записываются нули. Например, предыдущий регистр изменяется при сдвигах следующим образом:

$$\begin{aligned}
\mathtt{ShLo}: &\ \mathtt{645D4A58}_{16} = 01100100010111010100101001011000,\\
\mathtt{ShHi}: &\ \mathtt{91752962}_{16} = 10010001011101010010100101100010,\\
\mathtt{RotHi}: &\ \mathtt{91752963}_{16} = 10010001011101010010100101100011.
\end{aligned}$$

Выгружая из регистра октеты слева направо, получаем следующие результаты:

$$\begin{aligned} \texttt{ShLo}(w) &= \texttt{584A5D64}_{16}, \\ \texttt{ShHi}(w) &= \texttt{62297591}_{16}, \\ \texttt{RotHi}(w) &= \texttt{63297591}_{16}. \end{aligned}$$

Перестановки октетов при загрузке слова в регистр и при выгрузке из регистра в современных процессорах выполняются неявно.

При сдвигах на число позиций, кратное 8, операции ShLo, ShHi, RotHi интерпретируются намного проще и состоят в сдвиге октетов исходного слова: при ShLo — в сторону первых октетов, при ShHi — в сторону последних октетов, при RotHi — циклически в сторону последних октетов. Например,

$$\begin{aligned} \texttt{ShLo}^8(w) &= \texttt{94BAC800}_{16}, \\ \texttt{ShHi}^8(w) &= \texttt{00B194BA}_{16}, \\ \texttt{RotHi}^8(w) &= \texttt{C8B194BA}_{16}. \end{aligned}$$

4.2.3 Слова как многочлены

Октету $u = u_1 u_2 \dots u_8$ ставится в соответствие многочлен $u(x) = u_1 x^7 + u_2 x^6 + \dots + u_8$. Многочлен ставится в соответствие также любому непустому двоичному слову из целого числа октетов. Как и при представлении слов числами используется соглашение «от младших к старшим»: первому октету соответствует многочлен $u_1(x)$, второму — $x^8 u_2(x)$, третьему — $x^{16} u_3(x)$ и так далее.

Многочлены $u(x)$ считаются многочленами над полем $\mathbb{F}_2$. Это значит, что при сложении и умножении многочленов операции над их коэффициентами выполняются по модулю 2. Деление $u(x)$ на ненулевой $f(x)$ состоит в определении многочленов $q(x)$, $r(x)$ таких, что $u(x) = q(x)f(x) + r(x)$ и степень $r(x)$ меньше степени $f(x)$. Многочлен $r(x)$ является остатком от деления.

Операция $*$ состоит в умножении слов как многочленов с заменой результата умножения на его остаток от деления на $f(x) = x^{128} + x^7 + x^2 + x + 1$. Выбранный многочлен $f(x)$ является неприводимым (его нельзя представить в виде произведения многочленов меньших степеней). Поэтому операция $*$ задает умножение слов как элементов поля из 2^{128} элементов (подробнее см. [1]).

4.3 Запись перечислений

При записи последовательности $X_1, X_2, \dots, X_n$ допускается, если не оговорено противное, выполнение неравенства $n < 2$. При $n = 0$ определена пустая последовательность, а при $n = 1$ — одноэлементная последовательность X_1.

Аналогичные соглашения распространяются на запись разбиений слов, запись циклов и другие перечисления. Например, слово $X_1 \parallel X_2 \parallel \dots \parallel X_n$ является пустым при $n = 0$, тело цикла «для $i = 1, 2, \dots, n$ выполнить $\dots$» не выполняется ни разу, если $n = 0$, и выполняется один раз, если $n = 1$.

Для $X \in \{0,1\}^*$ запись

$$X = X_1 \parallel X_2 \parallel \dots \parallel X_n, \quad |X_1| = |X_2| = \dots = |X_{n-1}| = 128, \quad |X_n| \leqslant 128,$$

означает разбиение X слева направо на последовательные блоки до тех пор, пока они не будут исчерпаны и не будет определен последний, возможно неполный, блок X_n. Например, при $|X| = 128$ получаем разбиение с $n = 1$ и $|X_1| = 128$, при $|X| = 129$ — разбиение с $n = 2$ и $|X_2| = 1$. Если X — пустое слово, то $n = 0$ и блоки разбиения не определены.

5 Общие положения

5.1 Назначение

Настоящий стандарт определяет семейство криптографических алгоритмов, предназначенных для обеспечения конфиденциальности и контроля целостности данных. Обрабатываемыми данными являются двоичные слова (сообщения).

Криптографические алгоритмы стандарта построены на основе базовых алгоритмов шифрования блока данных. Базовые алгоритмы описываются в 6.1.

Криптографические алгоритмы шифрования и контроля целостности делятся на восемь групп:

1) алгоритмы шифрования в режиме простой замены (6.2);
2) алгоритмы шифрования в режиме сцепления блоков (6.3);
3) алгоритмы шифрования в режиме гаммирования с обратной связью (6.4);
4) алгоритмы шифрования в режиме счетчика (6.5);
5) алгоритм выработки имитовставки (6.6);
6) алгоритмы одновременного шифрования и имитозащиты данных (6.7);
7) алгоритмы одновременного шифрования и имитозащиты ключа (6.8);
8) алгоритм хэширования (6.9).

Первые четыре группы предназначены для обеспечения конфиденциальности сообщений. Каждая группа включает алгоритм зашифрования и алгоритм расшифрования. Стороны, располагающие общим ключом, могут организовать конфиденциальный обмен сообщениями путем их зашифрования перед отправкой и расшифрования после получения. В режимах простой замены и сцепления блоков шифруются сообщения, которые содержат хотя бы один блок, а в режимах гаммирования с обратной связью и счетчика — сообщения произвольной длины.

Пятый алгоритм предназначен для контроля целостности сообщений с помощью имитовставок — контрольных слов, которые определяются с использованием ключа. Стороны, располагающие общим ключом, могут организовать контроль целостности при обмене сообщениями путем добавления к ним имитовставок при отправке и проверки имитовставок при получении. Проверка имитовставок дополнительно позволяет стороне-получателю убедиться в том, что сторона-отправитель знает ключ, т. е. позволяет проверить подлинность сообщений.

Шестая и седьмая группы предназначены для обеспечения конфиденциальности и контроля целостности сообщений. Каждая группа включает алгоритмы установки и снятия защиты.

В шестой группе исходное сообщение задается двумя частями: открытой и критической. Алгоритмы защиты предназначены для контроля целостности обеих частей и обеспечения конфиденциальности критической части. При установке защиты вычисляет-

ся имитовставка всего сообщения и зашифровывается его критическая часть. При снятии защиты имитовставка проверяется и, если проверка прошла успешно, критическая часть расшифровывается.

В алгоритмах седьмой группы длина защищаемого сообщения должна быть сразу известна, эти алгоритмы рекомендуется применять для защиты ключей. Защищаемый ключ сопровождается открытым заголовком, который содержит открытые атрибуты ключа и одновременно является контрольным значением при проверке целостности. Могут использоваться фиксированные постоянные заголовки, которые служат только для контроля целостности. При установке защиты ключ зашифровывается вместе со своим заголовком и формируется слово, которое является одновременно защищенным ключом и имитовставкой ключа. При снятии защиты выполняется обратное преобразование и расшифрованный заголовок сравнивается с контрольным.

Восьмой алгоритм предназначен для вычисления хэш-значений — контрольных слов, которые определяются без использования ключа. Стороны могут организовать контроль целостности сообщений путем сравнения их хэш-значений с достоверными контрольными хэш-значениями. Изменение сообщения с высокой вероятностью приводит к изменению соответствующего хэш-значения и поэтому хэш-значения могут использоваться вместо самих сообщений, например в системах электронной цифровой подписи.

Дополнительно в разделе 7 определяются вспомогательные алгоритмы расширения и преобразования ключа, предназначенные для создания и модификации ключей шифрования и имитозащиты.

В приложении А приводятся примеры выполнения алгоритмов стандарта. Примеры можно использовать для проверки корректности реализаций алгоритмов.

В приложении Б приводится модуль абстрактно-синтаксической нотации версии 1 (АСН.1), определенной в ГОСТ 34.973. Модуль задает идентификаторы алгоритмов стандарта и описывает форматы параметров алгоритмов. Рекомендуется использовать модуль при встраивании алгоритмов в информационные системы, в которых также используется АСН.1.

5.2 Ключ

В алгоритмах шифрования и имитозащиты используется ключ $\theta \in \{0,1\}^{256}$. Ключ должен вырабатываться без возможности предсказания, распространяться с соблюдением мер конфиденциальности и храниться в секрете.

Разрешается использовать ключ θ, полученный в результате расширения короткого ключа длины 128 или 192. При этом должен использоваться алгоритм расширения, заданный в 7.1.

Один и тот же ключ не должен использоваться в алгоритмах различных групп.

В 7.2 определяется алгоритм преобразования ключа, с помощью которого по исходному ключу можно строить наборы новых ключей, которые, в свою очередь, также можно преобразовывать. Алгоритм преобразования может применяться для создания семейств ключей различного назначения, в том числе для использования в алгоритмах шифрования и имитозащиты различных групп. Кроме этого, алгоритм преобразования позволяет

организовать обновление ключей при исчерпании лимитов времени их использования или объема обработанных на ключах данных.

Ключам, которые требуется получить в результате преобразования, ставятся в соответствие заголовки $I \in \{0,1\}^{128}$, содержащие открытые атрибуты ключей, например, тип или назначение. Кроме этого, ключам назначаются уровни $D \in \{0,1\}^{96}$. Исходному ключу назначается уровень $\langle 0 \rangle_{96}$. Алгоритм преобразования по ключу уровня D и заголовку I строит новый ключ уровня $D \boxplus \langle 1 \rangle_{96}$ с заголовком I. Многократное применение алгоритма к одному ключу с различными заголовками I соответствует генерации семейства ключей различного назначения. Последовательное применение алгоритма к одному ключу с сохранением заголовка I соответствует обновлению ключа.

5.3 Синхропосылка

При шифровании в режимах сцепления блоков, гаммирования с обратной связью и счетчика, а также при одновременном шифровании и имитозащите данных используется синхропосылка $S \in \{0,1\}^{128}$.

Синхропосылка не является секретным параметром, может добавляться к зашифрованному сообщению и передаваться вместе с ним.

При шифровании в режимах гаммирования с обратной связью и счетчика, а также при одновременном шифровании и имитозащите данных должны использоваться уникальные синхропосылки. Уникальность означает, что при зашифровании или установке защиты на одном и том же ключе используются либо заведомо различные синхропосылки, либо вероятность совпадения синхропосылок пренебрежимо мала.

В режиме сцепления блоков синхропосылка должна быть не только уникальной, но и непредсказуемой. Непредсказуемость означает, что синхропосылки формируются случайно или по секретным правилам и вероятность угадать, какая синхропосылка будет использоваться, пренебрежимо мала.

Синхропосылки можно вырабатывать случайным или псевдослучайным методом, строить по меткам времени, значениям монотонного счетчика, неповторяющимся номерам сообщений и др. В режиме сцепления блоков предсказуемые значения не должны использоваться напрямую для построения синхропосылок, а должны предварительно зашифроваться на том же ключе, который используется для шифрования сообщений.

5.4 Имитовставка

В алгоритме выработки имитовставки и в алгоритмах одновременного шифрования и имитозащиты данных вычисляется либо проверяется имитовставка $T \in \{0,1\}^{64}$.

Если требуются не все, а $l < 64$ символов имитовставки, то должны использоваться первые l символов. При выборе l следует учитывать, что при навязывании ложного сообщения вероятность угадать с одной попытки его имитовставку, не зная ключ, равняется 2^{-l}.

5.5 Хэш-значение

В алгоритме хэширования вычисляется хэш-значение $Y \in \{0,1\}^{256}$.

Если требуются не все, а $l < 256$ символов хэш-значения, то должны использоваться первые l символов. При выборе l следует учитывать, что для определения сообщения с заданным хэш-значением требуется выполнить порядка 2^l операций, а для определения двух различных сообщений с одинаковыми хэш-значениями требуется выполнить порядка $2^{l/2}$ операций.

6 Алгоритмы шифрования и контроля целостности

6.1 Шифрование блока

6.1.1 Входные и выходные данные

Входными данными алгоритмов зашифрования и расшифрования являются блок $X \in \{0,1\}^{128}$ и ключ $\theta \in \{0,1\}^{256}$.

Выходными данными является блок $Y \in \{0,1\}^{128}$ — результат зашифрования либо расшифрования слова X на ключе θ: $Y = F_\theta(X)$ либо $Y = F_\theta^{-1}(X)$.

Входные данные подготавливаются следующим образом:

1 Слово X записывается в виде $X = X_1 \parallel X_2 \parallel X_3 \parallel X_4$, где $X_i \in \{0,1\}^{32}$.

2 Ключ θ записывается в виде $\theta = \theta_1 \parallel \theta_2 \parallel \ldots \parallel \theta_8$, $\theta_i \in \{0,1\}^{32}$, и определяются тактовые ключи $K_1 = \theta_1, K_2 = \theta_2, \ldots, K_8 = \theta_8, K_9 = \theta_1, K_{10} = \theta_2, \ldots, K_{56} = \theta_8$.

6.1.2 Вспомогательные преобразования и переменные

Подстановка H. Подстановка $H\colon \{0,1\}^8 \to \{0,1\}^8$ задается таблицей 2. В таблице 2 используется шестнадцатеричное представление слов $u \in \{0,1\}^8$. Если $u = \mathtt{IJ}_{16}$, то значение $H(u)$ находится на пересечении строки I и столбца J. Например, $H(\mathtt{A2}_{16}) = \mathtt{9B}_{16}$.

Таблица 2 — Подстановка H

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	B1	94	BA	C8	0A	08	F5	3B	36	6D	00	8E	58	4A	5D	E4
1	85	04	FA	9D	1B	B6	C7	AC	25	2E	72	C2	02	FD	CE	0D
2	5B	E3	D6	12	17	B9	61	81	FE	67	86	AD	71	6B	89	0B
3	5C	B0	C0	FF	33	C3	56	B8	35	C4	05	AE	D8	E0	7F	99
4	E1	2B	DC	1A	E2	82	57	EC	70	3F	CC	F0	95	EE	8D	F1
5	C1	AB	76	38	9F	E6	78	CA	F7	C6	F8	60	D5	BB	9C	4F
6	F3	3C	65	7B	63	7C	30	6A	DD	4E	A7	79	9E	B2	3D	31
7	3E	98	B5	6E	27	D3	BC	CF	59	1E	18	1F	4C	5A	B7	93
8	E9	DE	E7	2C	8F	0C	0F	A6	2D	DB	49	F4	6F	73	96	47
9	06	07	53	16	ED	24	7A	37	39	CB	A3	83	03	A9	8B	F6
A	92	BD	9B	1C	E5	D1	41	01	54	45	FB	C9	5E	4D	0E	F2
B	68	20	80	AA	22	7D	64	2F	26	87	F9	34	90	40	55	11
C	BE	32	97	13	43	FC	9A	48	A0	2A	88	5F	19	4B	09	A1
D	7E	CD	A4	D0	15	44	AF	8C	A5	84	50	BF	66	D2	E8	8A
E	A2	D7	46	52	42	A8	DF	B3	69	74	C5	51	EB	23	29	21
F	D4	EF	D9	B4	3A	62	28	75	91	14	10	EA	77	6C	DA	1D

Преобразования G_r $(r = 5, 13, 21)$. Преобразование $G_r\colon \{0,1\}^{32} \to \{0,1\}^{32}$ ставит в соответствие слову $u = u_1 \parallel u_2 \parallel u_3 \parallel u_4$, $u_i \in \{0,1\}^8$, слово

$$G_r(u) = \mathtt{RotHi}^r\left(H(u_1) \parallel H(u_2) \parallel H(u_3) \parallel H(u_4)\right).$$

Переменные. Используются переменные a, b, c, d, e со значениями из $\{0,1\}^{32}$.

6.1.3 Алгоритм зашифрования

Для зашифрования блока X на ключе θ выполняются следующие шаги:

1 Установить $a \leftarrow X_1$, $b \leftarrow X_2$, $c \leftarrow X_3$, $d \leftarrow X_4$.

2 Для $i = 1, 2, \ldots, 8$ выполнить (см. рисунок 1):

1) $b \leftarrow b \oplus G_5(a \boxplus K_{7i-6})$;
2) $c \leftarrow c \oplus G_{21}(d \boxplus K_{7i-5})$;
3) $a \leftarrow a \boxminus G_{13}(b \boxplus K_{7i-4})$;
4) $e \leftarrow G_{21}(b \boxplus c \boxplus K_{7i-3}) \oplus \langle i \rangle_{32}$;
5) $b \leftarrow b \boxplus e$;
6) $c \leftarrow c \boxminus e$;
7) $d \leftarrow d \boxplus G_{13}(c \boxplus K_{7i-2})$;
8) $b \leftarrow b \oplus G_{21}(a \boxplus K_{7i-1})$;
9) $c \leftarrow c \oplus G_5(d \boxplus K_{7i})$;
10) $a \leftrightarrow b$;
11) $c \leftrightarrow d$;
12) $b \leftrightarrow c$.

3 Установить $Y \leftarrow b \parallel d \parallel a \parallel c$.

4 Возвратить Y.

6.1.4 Алгоритм расшифрования

Для расшифрования блока X на ключе θ выполняются следующие шаги:

1 Установить $a \leftarrow X_1$, $b \leftarrow X_2$, $c \leftarrow X_3$, $d \leftarrow X_4$.

2 Для $i = 8, 7, \ldots, 1$ выполнить:

1) $b \leftarrow b \oplus G_5(a \boxplus K_{7i})$;
2) $c \leftarrow c \oplus G_{21}(d \boxplus K_{7i-1})$;
3) $a \leftarrow a \boxminus G_{13}(b \boxplus K_{7i-2})$;
4) $e \leftarrow G_{21}(b \boxplus c \boxplus K_{7i-3}) \oplus \langle i \rangle_{32}$;
5) $b \leftarrow b \boxplus e$;
6) $c \leftarrow c \boxminus e$;
7) $d \leftarrow d \boxplus G_{13}(c \boxplus K_{7i-4})$;
8) $b \leftarrow b \oplus G_{21}(a \boxplus K_{7i-5})$;
9) $c \leftarrow c \oplus G_5(d \boxplus K_{7i-6})$;
10) $a \leftrightarrow b$;
11) $c \leftrightarrow d$;
12) $a \leftrightarrow d$.

3 Установить $Y \leftarrow c \parallel a \parallel d \parallel b$.

4 Возвратить Y.

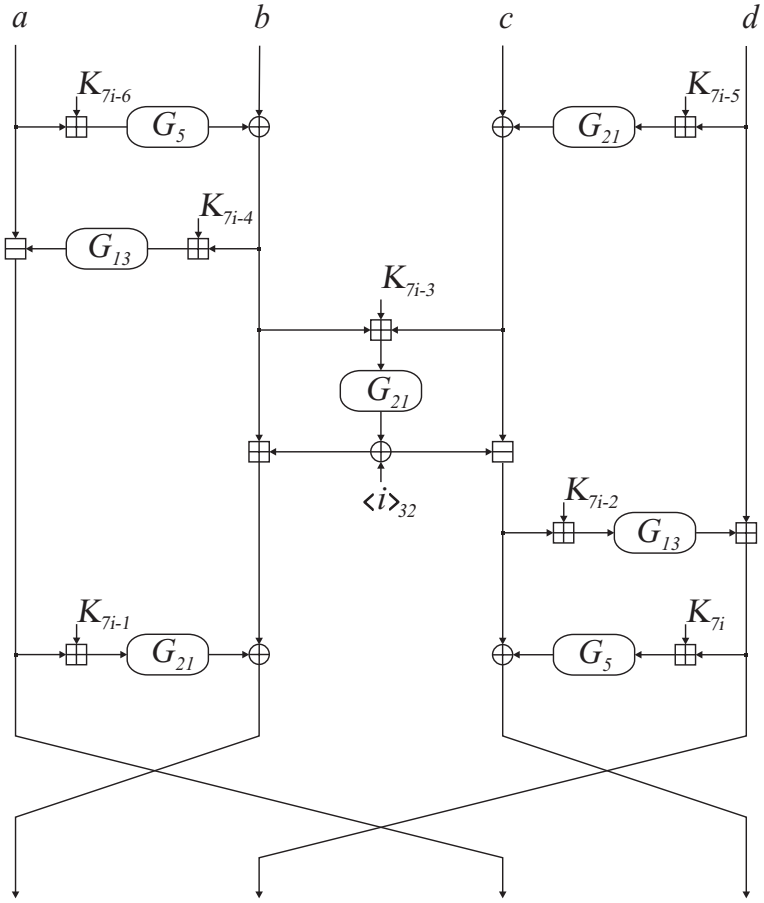

Рисунок 1 — Вычисления на i-м такте зашифрования

6.2 Шифрование в режиме простой замены

6.2.1 Входные и выходные данные

Входными данными алгоритмов зашифрования и расшифрования являются сообщение $X \in \{0,1\}^*$ и ключ $\theta \in \{0,1\}^{256}$. Длина X должна быть не меньше 128.

Выходными данными является слово $Y \in \{0,1\}^{|X|}$ — результат зашифрования либо расшифрования X на ключе θ.

Входное сообщение X записывается в виде

$$X = X_1 \parallel X_2 \parallel \ldots \parallel X_n, \quad |X_1| = |X_2| = \ldots = |X_{n-1}| = 128, \quad 0 < |X_n| \leqslant 128.$$

При шифровании словам X_i ставятся в соответствие слова $Y_i \in \{0,1\}^{|X_i|}$, из которых затем составляется Y.

6.2.2 Переменные

При $|X_n| < 128$ используется переменная r со значениями из $\{0,1\}^{128-|X_n|}$.

6.2.3 Алгоритм зашифрования

Зашифрование сообщения X на ключе θ состоит в выполнении следующих шагов:

1 Если $|X_n| = 128$, то

1) для $i = 1, 2, \ldots, n$ выполнить: $Y_i \leftarrow F_\theta(X_i)$.

2 Если $|X_n| < 128$, то

1) для $i = 1, 2, \ldots, n-2$ выполнить: $Y_i \leftarrow F_\theta(X_i)$;
2) $Y_n \parallel r \leftarrow F_\theta(X_{n-1})$;
3) $Y_{n-1} \leftarrow F_\theta(X_n \parallel r)$.

3 Установить $Y \leftarrow Y_1 \parallel Y_2 \parallel \ldots \parallel Y_n$.

4 Возвратить Y.

6.2.4 Алгоритм расшифрования

Расшифрование сообщения X на ключе θ состоит в выполнении следующих шагов:

1 Если $|X_n| = 128$, то
1) для $i = 1, 2, \ldots, n$ выполнить: $Y_i \leftarrow F_\theta^{-1}(X_i)$.

2 Если $|X_n| < 128$, то
1) для $i = 1, 2, \ldots, n-2$ выполнить: $Y_i \leftarrow F_\theta^{-1}(X_i)$;
2) $Y_n \parallel r \leftarrow F_\theta^{-1}(X_{n-1})$;
3) $Y_{n-1} \leftarrow F_\theta^{-1}(X_n \parallel r)$.

3 Установить $Y \leftarrow Y_1 \parallel Y_2 \parallel \ldots \parallel Y_n$.

4 Возвратить Y.

6.3 Шифрование в режиме сцепления блоков

6.3.1 Входные и выходные данные

Входными данными алгоритмов зашифрования и расшифрования являются сообщение $X \in \{0,1\}^*$, ключ $\theta \in \{0,1\}^{256}$ и синхропосылка $S \in \{0,1\}^{128}$. Длина X должна быть не меньше 128.

Выходными данными является слово $Y \in \{0,1\}^{|X|}$ — результат зашифрования либо расшифрования X на ключе θ при использовании синхропосылки S.

Входное сообщение X записывается в виде

$$X = X_1 \parallel X_2 \parallel \ldots \parallel X_n, \quad |X_1| = |X_2| = \ldots = |X_{n-1}| = 128, \quad 0 < |X_n| \leqslant 128.$$

При шифровании словам X_i ставятся в соответствие слова $Y_i \in \{0,1\}^{|X_i|}$, из которых затем составляется Y.

При зашифровании используется вспомогательный блок $Y_0 \in \{0,1\}^{128}$, а при расшифровании — вспомогательный блок $X_0 \in \{0,1\}^{128}$.

6.3.2 Переменные

При $|X_n| < 128$ используется переменная r со значениями из $\{0,1\}^{128-|X_n|}$.

6.3.3 Алгоритм зашифрования

Зашифрование сообщения X на ключе θ при использовании синхропосылки S состоит в выполнении следующих шагов:

1 Установить $Y_0 \leftarrow S$.

2 Если $|X_n| = 128$, то
1) для $i = 1, 2, \ldots, n$ выполнить: $Y_i \leftarrow F_\theta(X_i \oplus Y_{i-1})$.

3 Если $|X_n| < 128$, то
1) для $i = 1, 2, \ldots, n-2$ выполнить: $Y_i \leftarrow F_\theta(X_i \oplus Y_{i-1})$;

2) $Y_n \parallel r \leftarrow F_\theta(X_{n-1} \oplus Y_{n-2})$;
3) $Y_{n-1} \leftarrow F_\theta((X_n \oplus Y_n) \parallel r)$.

4 Установить $Y \leftarrow Y_1 \parallel Y_2 \parallel \ldots \parallel Y_n$.
5 Возвратить Y.

6.3.4 Алгоритм расшифрования

Расшифрование сообщения X на ключе θ при использовании синхропосылки S состоит в выполнении следующих шагов:

1 Установить $X_0 \leftarrow S$.
2 Если $|X_n| = 128$, то
1) для $i = 1, 2, \ldots, n$ выполнить: $Y_i \leftarrow F_\theta^{-1}(X_i) \oplus X_{i-1}$.
3 Если $|X_n| < 128$, то
1) для $i = 1, 2, \ldots, n-2$ выполнить: $Y_i \leftarrow F_\theta^{-1}(X_i) \oplus X_{i-1}$;
2) $Y_n \parallel r \leftarrow F_\theta^{-1}(X_{n-1}) \oplus (X_n \parallel 0^{128-|X_n|})$;
3) $Y_{n-1} \leftarrow F_\theta^{-1}(X_n \parallel r) \oplus X_{n-2}$.
4 Установить $Y \leftarrow Y_1 \parallel Y_2 \parallel \ldots \parallel Y_n$.
5 Возвратить Y.

6.4 Шифрование в режиме гаммирования с обратной связью

6.4.1 Входные и выходные данные

Входными данными алгоритмов зашифрования и расшифрования являются сообщение $X \in \{0,1\}^*$, ключ $\theta \in \{0,1\}^{256}$ и синхропосылка $S \in \{0,1\}^{128}$.

Выходными данными является слово $Y \in \{0,1\}^{|X|}$ — результат зашифрования либо расшифрования X на ключе θ при использовании синхропосылки S.

Входное сообщение X записывается в виде

$$X = X_1 \parallel X_2 \parallel \ldots \parallel X_n, \quad |X_1| = |X_2| = \ldots = |X_{n-1}| = 128, \quad |X_n| \leqslant 128.$$

При шифровании словам X_i ставятся в соответствие слова $Y_i \in \{0,1\}^{|X_i|}$, из которых затем составляется Y.

При зашифровании используется вспомогательный блок $Y_0 \in \{0,1\}^{128}$, а при расшифровании — вспомогательный блок $X_0 \in \{0,1\}^{128}$.

6.4.2 Алгоритм зашифрования

Зашифрование сообщения X на ключе θ при использовании синхропосылки S состоит в выполнении следующих шагов:

1 Установить $Y_0 \leftarrow S$.
2 Для $i = 1, 2, \ldots, n$ выполнить: $Y_i \leftarrow X_i \oplus L_{|X_i|}(F_\theta(Y_{i-1}))$.
3 Установить $Y \leftarrow Y_1 \parallel Y_2 \parallel \ldots \parallel Y_n$.
4 Возвратить Y.

6.4.3 Алгоритм расшифрования

Расшифрование сообщения X на ключе θ при использовании синхропосылки S состоит в выполнении следующих шагов:

1 Установить $X_0 \leftarrow S$.
2 Для $i = 1, 2, \ldots, n$ выполнить: $Y_i \leftarrow X_i \oplus L_{|X_i|}(F_\theta(X_{i-1}))$.
3 Установить $Y \leftarrow Y_1 \| Y_2 \| \ldots \| Y_n$.
4 Возвратить Y.

6.5 Шифрование в режиме счетчика

6.5.1 Входные и выходные данные

Входными данными алгоритмов зашифрования и расшифрования являются сообщение $X \in \{0,1\}^*$, ключ $\theta \in \{0,1\}^{256}$ и синхропосылка $S \in \{0,1\}^{128}$.

Выходными данными является слово $Y \in \{0,1\}^{|X|}$ — результат зашифрования либо расшифрования X на ключе θ при использовании синхропосылки S.

Входное сообщение X записывается в виде

$$X = X_1 \| X_2 \| \ldots \| X_n, \quad |X_1| = |X_2| = \ldots = |X_{n-1}| = 128, \quad |X_n| \leqslant 128.$$

При шифровании словам X_i ставятся в соответствие слова $Y_i \in \{0,1\}^{|X_i|}$, из которых затем составляется Y.

6.5.2 Переменные

Используется переменная s со значениями из $\{0,1\}^{128}$.

6.5.3 Алгоритм зашифрования

Зашифрование сообщения X на ключе θ при использовании синхропосылки S состоит в выполнении следующих шагов:

1 Установить $s \leftarrow F_\theta(S)$.
2 Для $i = 1, 2, \ldots, n$ выполнить:
  1) $s \leftarrow s \boxplus \langle 1 \rangle_{128}$,
  2) $Y_i \leftarrow X_i \oplus L_{|X_i|}(F_\theta(s))$.
3 Установить $Y \leftarrow Y_1 \| Y_2 \| \ldots \| Y_n$.
4 Возвратить Y.

6.5.4 Алгоритм расшифрования

Расшифрование сообщения X на ключе θ при использовании синхропосылки S состоит в выполнении тех же шагов, что и при зашифровании.

6.6 Выработка имитовставки

6.6.1 Входные и выходные данные

Входными данными алгоритма выработки имитовставки являются сообщение $X \in \{0,1\}^*$ и ключ $\theta \in \{0,1\}^{256}$.

Выходными данными является слово $T \in \{0,1\}^{64}$ — имитовставка X на ключе θ.

Входное сообщение X ненулевой длины записывается в виде

$$X = X_1 \| X_2 \| \ldots \| X_n, \quad |X_1| = |X_2| = \ldots = |X_{n-1}| = 128, \quad 0 < |X_n| \leqslant 128.$$

Если же X — пустое слово, то $n = 1$ и $|X_1| = 0$.

6.6.2 Вспомогательные преобразования и переменные

Преобразования φ_1 и φ_2. Преобразования $\varphi_1, \varphi_2\colon \{0,1\}^{128} \to \{0,1\}^{128}$ действуют на слово $u = u_1 \parallel u_2 \parallel u_3 \parallel u_4$, $u_i \in \{0,1\}^{32}$, по правилам:

$$\varphi_1(u) = u_2 \parallel u_3 \parallel u_4 \parallel (u_1 \oplus u_2),$$
$$\varphi_2(u) = (u_1 \oplus u_4) \parallel u_1 \parallel u_2 \parallel u_3.$$

Отображение ψ. Отображение ψ ставит в соответствие двоичному слову u, длина которого меньше 128, слово $\psi(u) = u \parallel 1 \parallel 0^{127-|u|}$ длины 128.

Переменные. Используются переменные r и s со значениями из $\{0,1\}^{128}$.

6.6.3 Алгоритм выработки имитовставки

Определение имитовставки сообщения X на ключе θ состоит в выполнении следующих шагов:

1 Установить $s \leftarrow 0^{128}$, $r \leftarrow F_\theta(s)$.

2 Для $i = 1, 2, \ldots, n-1$ выполнить: $s \leftarrow F_\theta(s \oplus X_i)$.

3 Если $|X_n| = 128$, то $s \leftarrow s \oplus X_n \oplus \varphi_1(r)$, иначе $s \leftarrow s \oplus \psi(X_n) \oplus \varphi_2(r)$.

4 Установить $T \leftarrow L_{64}(F_\theta(s))$.

5 Возвратить T.

6.7 Шифрование и имитозащита данных

6.7.1 Входные и выходные данные

Входными данными алгоритма установки защиты являются критическое сообщение $X \in \{0,1\}^*$, открытое сообщение $I \in \{0,1\}^*$, ключ $\theta \in \{0,1\}^{256}$ и синхропосылка $S \in \{0,1\}^{128}$. Длины I и X должны быть меньше 2^{64}.

Выходными данными алгоритма установки защиты являются слово $Y \in \{0,1\}^{|X|}$ — результат зашифрования X на ключе θ при использовании синхропосылки S, и слово $T \in \{0,1\}^{64}$ — имитовставка пары (X, I) на ключе θ при использовании синхропосылки S.

Входными данными алгоритма снятия защиты являются зашифрованное сообщение $X \in \{0,1\}^*$, открытое сообщение $I \in \{0,1\}^*$, имитовставка $T \in \{0,1\}^{64}$, ключ $\theta \in \{0,1\}^{256}$ и синхропосылка $S \in \{0,1\}^{128}$. Длины X и I должны быть меньше 2^{64}.

Выходными данными алгоритма снятия защиты является признак `ОШИБКА` либо слово $Y \in \{0,1\}^{|X|}$ — результат расшифрования X на ключе θ при использовании синхропосылки S. Возврат признака `ОШИБКА` означает нарушение целостности данных.

Входные сообщения X и I записываются в виде

$$X = X_1 \parallel X_2 \parallel \ldots \parallel X_n, \qquad |X_1| = |X_2| = \ldots = |X_{n-1}| = 128, \quad |X_n| \leqslant 128,$$
$$I = I_1 \parallel I_2 \parallel \ldots \parallel I_m, \qquad |I_1| = |I_2| = \ldots = |I_{m-1}| = 128, \quad |I_m| \leqslant 128.$$

При шифровании словам X_i ставятся в соответствие слова $Y_i \in \{0,1\}^{|X_i|}$, из которых затем составляется Y.

6.7.2 Переменные

Используются переменные r и s со значениями из $\{0,1\}^{128}$.

6.7.3 Алгоритм установки защиты

Защита пары (X, I) на ключе θ при использовании синхропосылки S состоит в выполнении следующих шагов:

1 Установить $r \leftarrow F_\theta(S)$, $s \leftarrow r$.

2 Для $i = 1, 2, \ldots, n$ выполнить:

1) $s \leftarrow s \boxplus \langle 1 \rangle_{128}$;

2) $Y_i \leftarrow X_i \oplus L_{|X_i|}(F_\theta(s))$.

3 Установить $r \leftarrow F_\theta(r)$, $s \leftarrow \mathtt{B194BAC80A08F53B366D008E584A5DE4}_{16}$, где последнее присваиваемое значение определяется последовательными элементами первой строки таблицы 2.

4 Для $i = 1, 2, \ldots, m$ выполнить:

1) $s \leftarrow s \oplus (I_i \parallel 0^{128-|I_i|})$;

2) $s \leftarrow s * r$.

5 Для $i = 1, 2, \ldots, n$ выполнить:

1) $s \leftarrow s \oplus (Y_i \parallel 0^{128-|Y_i|})$;

2) $s \leftarrow s * r$.

6 Установить $s \leftarrow s \oplus (\langle |I| \rangle_{64} \parallel \langle |X| \rangle_{64})$;

7 Установить $s \leftarrow F_\theta(s * r)$.

8 Установить $T \leftarrow L_{64}(s)$.

9 Возвратить (Y, T).

6.7.4 Алгоритм снятия защиты

Снятие защиты с тройки (X, I, T) на ключе θ при использовании синхропосылки S состоит в выполнении следующих шагов:

1 Установить $r \leftarrow F_\theta(F_\theta(S))$, $s \leftarrow \mathtt{B194BAC80A08F53B366D008E584A5DE4}_{16}$, где последнее присваиваемое значение определяется последовательными элементами первой строки таблицы 2.

2 Для $i = 1, 2, \ldots, m$ выполнить:

1) $s \leftarrow s \oplus (I_i \parallel 0^{128-|I_i|})$;

2) $s \leftarrow s * r$.

3 Для $i = 1, 2, \ldots, n$ выполнить:

1) $s \leftarrow s \oplus (X_i \parallel 0^{128-|X_i|})$;

2) $s \leftarrow s * r$.

4 Установить $s \leftarrow s \oplus (\langle |I| \rangle_{64} \parallel \langle |X| \rangle_{64})$;

5 Установить $s \leftarrow F_\theta(s * r)$.

6 Если $T \neq L_{64}(s)$, то возвратить `ОШИБКА`.

7 Установить $s \leftarrow F_\theta(S)$.

8 Для $i = 1, 2, \ldots, n$ выполнить:

1) $s \leftarrow s \boxplus \langle 1 \rangle_{128}$;

2) $Y_i \leftarrow X_i \oplus L_{|X_i|}(F_\theta(s))$.

9 Возвратить Y.

6.8 Шифрование и имитозащита ключа

6.8.1 Входные и выходные данные

Входными данными алгоритма установки защиты являются защищаемый ключ $X \in \{0,1\}^{8*}$, его заголовок $I \in \{0,1\}^{128}$ и ключ защиты $\theta \in \{0,1\}^{256}$. Длина X должна быть не меньше 128.

Выходными данными алгоритма установки защиты является слово $Y \in \{0,1\}^{|X|+128}$ — результат защиты пары (X, I) на ключе θ.

Входными данными алгоритма снятия защиты являются защищенный ключ $X \in \{0,1\}^{*}$, его заголовок $I \in \{0,1\}^{128}$ и ключ защиты $\theta \in \{0,1\}^{256}$.

Выходными данными алгоритма снятия защиты является признак `ОШИБКА` либо слово $Y \in \{0,1\}^{|X|-128}$ — результат снятия защиты с пары (X, I) на ключе θ. Возврат признака `ОШИБКА` означает нарушение целостности входных данных.

По входным данным алгоритмов определяется количество блоков

$$n = \begin{cases} \lceil |X|/128 \rceil + 1 & \text{при установке защиты,} \\ \lceil |X|/128 \rceil & \text{при снятии защиты.} \end{cases}$$

6.8.2 Переменные

Переменная r. Используется переменная r, которая при установке защиты принимает значения из $\{0,1\}^{|X|+128}$, а при снятии защиты — значения из $\{0,1\}^{|X|}$.

Значение r записывается в виде

$$r = r_1 \parallel r_2 \parallel \ldots \parallel r_{n-1} \parallel r_n, \quad |r_1| = |r_2| = \ldots = |r_{n-1}| = 128, \quad 0 < |r_n| \leqslant 128,$$

либо в виде

$$r = r^{**} \parallel r^*, \quad |r^*| = 128.$$

Переменная s. Используется переменная s со значениями из $\{0,1\}^{128}$.

6.8.3 Алгоритм установки защиты

Защита пары (X, I) на ключе θ состоит в выполнении следующих шагов:

1 Установить $r \leftarrow X \parallel I$.

2 Для $i = 1, 2, \ldots, 2n$ выполнить:

  1) $s \leftarrow r_1 \oplus r_2 \oplus \ldots \oplus r_{n-1}$;

  2) $r^* \leftarrow r^* \oplus F_\theta(s) \oplus \langle i \rangle_{128}$;

  3) $r \leftarrow \mathtt{ShLo}^{128}(r)$;

  4) $r^* \leftarrow s$.

3 Установить $Y \leftarrow r$.

4 Возвратить Y.

6.8.4 Алгоритм снятия защиты

Снятие защиты с пары (X, I) на ключе θ состоит в выполнении следующих шагов:

1 Если длина X не кратна 8 или $|X| < 256$, то возвратить `ОШИБКА`.

2 Установить $r \leftarrow X$.

3 Для $i = 2n, \ldots, 2, 1$ выполнить:

1) $s \leftarrow r^*$;
2) $r \leftarrow \mathtt{ShHi}^{128}(r)$;
3) $r^* \leftarrow r^* \oplus F_\theta(s) \oplus \langle i \rangle_{128}$;
4) $r_1 \leftarrow s \oplus r_2 \oplus \ldots \oplus r_{n-1}$.

4 Если $r^* \neq I$, то возвратить ОШИБКА.

5 Установить $Y \leftarrow r^{**}$.

6 Возвратить Y.

6.9 Хэширование

6.9.1 Входные и выходные данные

Входными данными алгоритма хэширования является сообщение $X \in \{0,1\}^*$.

Выходными данными является слово $Y \in \{0,1\}^{256}$ — хэш-значение сообщения X.

К входному сообщению X предварительно добавляется t нулевых символов, где t — минимальное неотрицательное целое число такое, что $|X|+t$ кратно 256. Полученное слово записывается в виде

$$X \parallel 0^t = X_1 \parallel X_2 \parallel \ldots \parallel X_n, \quad |X_1| = |X_2| = \ldots = |X_n| = 256.$$

6.9.2 Вспомогательные преобразования и переменные

Отображения σ_1 и σ_2. Отображения $\sigma_1\colon \{0,1\}^{512} \to \{0,1\}^{128}$ и $\sigma_2\colon \{0,1\}^{512} \to \{0,1\}^{256}$ действуют на слово $u = u_1 \parallel u_2 \parallel u_3 \parallel u_4$, $u_i \in \{0,1\}^{128}$, по правилам:

$$\sigma_1(u) = F_{u_1 \parallel u_2}(u_3 \oplus u_4) \oplus u_3 \oplus u_4,$$
$$\sigma_2(u) = (F_{\theta_1}(u_1) \oplus u_1) \parallel (F_{\theta_2}(u_2) \oplus u_2),$$

где $\theta_1 = \sigma_1(u) \parallel u_4$, $\theta_2 = (\sigma_1(u) \oplus 1^{128}) \parallel u_3$.

Переменные. Используется переменная s со значениями из $\{0,1\}^{128}$ и переменная h со значениями из $\{0,1\}^{256}$.

6.9.3 Алгоритм хэширования

Хэширование сообщения X состоит в выполнении следующих шагов:

1 Установить $s \leftarrow 0^{128}$.

2 Установить

$$h \leftarrow \mathtt{B194BAC80A08F53B366D008E584A5DE48504FA9D1BB6C7AC252E72C202FDCE0D}_{16},$$

где присваиваемое значение определяется последовательными (слева направо и сверху вниз) элементами первых двух строк таблицы 2.

3 Для $i = 1, 2, \ldots, n$ выполнить:
1) $s \leftarrow s \oplus \sigma_1(X_i \parallel h)$,
2) $h \leftarrow \sigma_2(X_i \parallel h)$.

4 Установить $Y \leftarrow \sigma_2(\langle |X| \rangle_{128} \parallel s \parallel h)$.

5 Возвратить Y.

7 Вспомогательные алгоритмы

7.1 Расширение ключа

7.1.1 Входные и выходные данные

Входными данными алгоритма расширения является исходный ключ $\theta_1 \| \theta_2 \| \ldots \| \theta_n$, где $\theta_i \in \{0,1\}^{32}$, $n \in \{4,6,8\}$.

Выходными данными алгоритма является ключ $\theta \in \{0,1\}^{256}$.

7.1.2 Алгоритм расширения ключа

Расширение ключа $\theta_1 \| \theta_2 \| \ldots \| \theta_n$ состоит в дополнении его словами $\theta_{n+1}, \ldots, \theta_8 \in \{0,1\}^{32}$ по следующим правилам:

1 Если $n = 4$, то выполнить:
  1) $\theta_5 \leftarrow \theta_1$;
  2) $\theta_6 \leftarrow \theta_2$;
  3) $\theta_7 \leftarrow \theta_3$;
  4) $\theta_8 \leftarrow \theta_4$.

2 Если $n = 6$, то выполнить:
  1) $\theta_7 \leftarrow \theta_1 \oplus \theta_2 \oplus \theta_3$;
  2) $\theta_8 \leftarrow \theta_4 \oplus \theta_5 \oplus \theta_6$.

3 Установить $\theta \leftarrow \theta_1 \| \theta_2 \| \ldots \| \theta_8$.

4 Возвратить θ.

7.2 Преобразование ключа

7.2.1 Входные и выходные данные

Входными данными алгоритма преобразования являются преобразуемый ключ $X \in \{0,1\}^n$, где $n \in \{128, 192, 256\}$, его уровень $D \in \{0,1\}^{96}$, заголовок $I \in \{0,1\}^{128}$ нового ключа и его длина $m \in \{128, 192, 256\}$, $m \leqslant n$.

Выходными данными алгоритма преобразования является слово $Y \in \{0,1\}^m$ — ключ с заголовком I, полученный по ключу X уровня D.

При преобразовании ключа Y его уровень должен полагаться равным $D \boxplus \langle 1 \rangle_{96}$.

7.2.2 Вспомогательные преобразования и переменные

Отображение σ_2. Используется отображение $\sigma_2 \colon \{0,1\}^{512} \to \{0,1\}^{256}$, действие которого определяется в соответствии с 6.9.2.

Алгоритм `KeyExpand`. Используется алгоритм расширения ключа `KeyExpand`, определенный в 7.1.2.

Переменные. Используется переменная r со значениями из $\{0,1\}^{32}$ и переменная s со значениями из $\{0,1\}^{256}$.

7.2.3 Алгоритм преобразования ключа

Преобразование ключа X уровня D на заголовке I состоит в выполнении следующих шагов:

1 Присвоить переменной r значение:
   1) $\texttt{B194BAC8}_{16}$, если $n = m = 128$;
   2) $\texttt{5BE3D612}_{16}$, если $n = 192$ и $m = 128$;
   3) $\texttt{5CB0C0FF}_{16}$, если $n = m = 192$;
   4) $\texttt{E12BDC1A}_{16}$, если $n = 256$ и $m = 128$;
   5) $\texttt{C1AB7638}_{16}$, если $n = 256$ и $m = 192$;
   6) $\texttt{F33C657B}_{16}$, если $n = m = 256$.
2 Установить $s \leftarrow \texttt{KeyExpand}(X)$.
3 Установить $Y \leftarrow L_m(\sigma_2(r \parallel D \parallel I \parallel s))$.
4 Возвратить Y.

Приложение А
(справочное)
Тестовые примеры

А.1 Шифрование блока

В таблице А.1 представлен пример зашифрования блока. Значения переменных a, b, c, d после выполнения тактов зашифрования указаны в таблице А.2. Значения переменных a, b, c, d после выполнения шагов первого такта зашифрования представлены в таблице А.3. Дополнительная переменная e после выполнения шага 4) принимает значение $\texttt{20072EC1}_{16}$.

Таблица А.1 — Зашифрование блока

X	$\texttt{B194BAC8 0A08F53B 366D008E 584A5DE4}_{16}$
θ	$\texttt{E9DEE72C 8F0C0FA6 2DDB49F4 6F739647 06075316 ED247A37 39CBA383 03A98BF6}_{16}$
Y	$\texttt{69CCA1C9 3557C9E3 D66BC3E0 FA88FA6E}_{16}$

Таблица А.2 — Такты зашифрования

Номер такта i	Переменные			
	a	b	c	d
1	$\texttt{FB56C62C}_{16}$	$\texttt{CA8EEEB7}_{16}$	$\texttt{09BAD702}_{16}$	$\texttt{CC4E441D}_{16}$
2	$7280\texttt{A094}_{16}$	$\texttt{47BB9CD6}_{16}$	$\texttt{5BD130B1}_{16}$	$\texttt{ADA525A4}_{16}$
3	$\texttt{00AB0E4D}_{16}$	$4\texttt{B4A6113}_{16}$	$73\texttt{D9CD18}_{16}$	$57\texttt{E54345}_{16}$
4	$\texttt{A50D12EF}_{16}$	$\texttt{8CD05085}_{16}$	$\texttt{99A672B7}_{16}$	$\texttt{D9A0C0E4}_{16}$
5	$21\texttt{C32063}_{16}$	$\texttt{44712C59}_{16}$	$\texttt{EC21160A}_{16}$	$\texttt{DE08AAB9}_{16}$
6	$\texttt{B5279D32}_{16}$	$\texttt{D4579966}_{16}$	$251\texttt{E3B2D}_{16}$	$\texttt{F8EF6A0F}_{16}$
7	26349022_{16}	$08\texttt{C5172E}_{16}$	$705\texttt{A63C6}_{16}$	$5\texttt{CA6AD61}_{16}$
8	$\texttt{D66BC3E0}_{16}$	$\texttt{69CCA1C9}_{16}$	$\texttt{FA88FA6E}_{16}$	$3557\texttt{C9E3}_{16}$

Таблица А.3 — Первый такт зашифрования

Шаг вычислений	Переменные			
	a	b	c	d
1) $b \leftarrow b \oplus G_5(a \boxplus K_1)$	$\mathtt{B194BAC8}_{16}$	$\mathtt{66DC9868}_{16}$	$\mathtt{366D008E}_{16}$	$\mathtt{584A5DE4}_{16}$
2) $c \leftarrow c \oplus G_{21}(d \boxplus K_2)$	$\mathtt{B194BAC8}_{16}$	$\mathtt{66DC9868}_{16}$	$\mathtt{F95E6998}_{16}$	$\mathtt{584A5DE4}_{16}$
3) $a \leftarrow a \boxminus G_{13}(b \boxplus K_3)$	$\mathtt{09BAD702}_{16}$	$\mathtt{66DC9868}_{16}$	$\mathtt{F95E6998}_{16}$	$\mathtt{584A5DE4}_{16}$
4) $e \leftarrow G_{21}(b \boxplus c \boxplus K_4) \oplus \langle 1 \rangle_{32}$	$\mathtt{09BAD702}_{16}$	$\mathtt{66DC9868}_{16}$	$\mathtt{F95E6998}_{16}$	$\mathtt{584A5DE4}_{16}$
5) $b \leftarrow b \boxplus e$	$\mathtt{09BAD702}_{16}$	$\mathtt{86E3C629}_{16}$	$\mathtt{F95E6998}_{16}$	$\mathtt{584A5DE4}_{16}$
6) $c \leftarrow c \boxminus e$	$\mathtt{09BAD702}_{16}$	$\mathtt{86E3C629}_{16}$	$\mathtt{D9573BD7}_{16}$	$\mathtt{584A5DE4}_{16}$
7) $d \leftarrow d \boxplus G_{13}(c \boxplus K_5)$	$\mathtt{09BAD702}_{16}$	$\mathtt{86E3C629}_{16}$	$\mathtt{D9573BD7}_{16}$	$\mathtt{CA8EEEB7}_{16}$
8) $b \leftarrow b \oplus G_{21}(a \boxplus K_6)$	$\mathtt{09BAD702}_{16}$	$\mathtt{FB56C62C}_{16}$	$\mathtt{D9573BD7}_{16}$	$\mathtt{CA8EEEB7}_{16}$
9) $c \leftarrow c \oplus G_5(d \boxplus K_7)$	$\mathtt{09BAD702}_{16}$	$\mathtt{FB56C62C}_{16}$	$\mathtt{CC4E441D}_{16}$	$\mathtt{CA8EEEB7}_{16}$
10) $a \leftrightarrow b$	$\mathtt{FB56C62C}_{16}$	$\mathtt{09BAD702}_{16}$	$\mathtt{CC4E441D}_{16}$	$\mathtt{CA8EEEB7}_{16}$
11) $c \leftrightarrow d$	$\mathtt{FB56C62C}_{16}$	$\mathtt{09BAD702}_{16}$	$\mathtt{CA8EEEB7}_{16}$	$\mathtt{CC4E441D}_{16}$
12) $b \leftrightarrow c$	$\mathtt{FB56C62C}_{16}$	$\mathtt{CA8EEEB7}_{16}$	$\mathtt{09BAD702}_{16}$	$\mathtt{CC4E441D}_{16}$

В таблице А.4 представлен пример расшифрования блока. Значения переменных a, b, c, d после выполнения тактов расшифрования указаны в таблице А.5.

Таблица А.4 — Расшифрование блока

X	$\mathtt{E12BDC1A\ E28257EC\ 703FCCF0\ 95EE8DF1}_{16}$
θ	$\mathtt{92BD9B1C\ E5D14101\ 5445FBC9\ 5E4D0EF2\ 682080AA\ 227D642F\ 2687F934\ 90405511}_{16}$
Y	$\mathtt{0DC53006\ 00CAB840\ B38448E5\ E993F421}_{16}$

Таблица А.5 — Такты расшифрования

Номер такта i	Переменные			
	a	b	c	d
8	$\mathtt{A174D6FC}_{16}$	$\mathtt{377EB086}_{16}$	$\mathtt{BA7C2D07}_{16}$	$\mathtt{0DAA044B}_{16}$
7	$\mathtt{B01E75B3}_{16}$	$\mathtt{0F53A46F}_{16}$	$\mathtt{8893A01F}_{16}$	$\mathtt{A4E35989}_{16}$
6	$\mathtt{B5B85383}_{16}$	$\mathtt{33D8BC0E}_{16}$	$\mathtt{9A46CD5F}_{16}$	$\mathtt{F8D778D4}_{16}$
5	$\mathtt{07234634}_{16}$	$\mathtt{723B48FC}_{16}$	$\mathtt{04690666}_{16}$	$\mathtt{ADB565F3}_{16}$
4	$\mathtt{3141A829}_{16}$	$\mathtt{2AD3FB40}_{16}$	$\mathtt{D30032B1}_{16}$	$\mathtt{4D336185}_{16}$
3	$\mathtt{ADA2EC35}_{16}$	$\mathtt{DADBC720}_{16}$	$\mathtt{3421AC22}_{16}$	$\mathtt{22EC7943}_{16}$
2	$\mathtt{9DAC9289}_{16}$	$\mathtt{89A2E5ED}_{16}$	$\mathtt{9253A0F0}_{16}$	$\mathtt{3B871FA3}_{16}$
1	$\mathtt{00CAB840}_{16}$	$\mathtt{E993F421}_{16}$	$\mathtt{0DC53006}_{16}$	$\mathtt{B38448E5}_{16}$

А.2 Шифрование в режиме простой замены

В таблицах А.6, А.7 представлены примеры зашифрования в режиме простой замены. В таблицах А.8, А.9 представлены примеры расшифрования в этом же режиме.

Таблица А.6 — Зашифрование в режиме простой замены ($|X| = 384$)

X	B194BAC8 0A08F53B 366D008E 584A5DE4 8504FA9D 1BB6C7AC 252E72C2 02FDCE0D 5BE3D612 17B96181 FE6786AD 716B890B$_{16}$
θ	E9DEE72C 8F0C0FA6 2DDB49F4 6F739647 06075316 ED247A37 39CBA383 03A98BF6$_{16}$
Y	69CCA1C9 3557C9E3 D66BC3E0 FA88FA6E 5F23102E F1097107 75017F73 806DA9DC 46FB2ED2 CE771F26 DCB5E5D1 569F9AB0$_{16}$

Таблица А.7 — Зашифрование в режиме простой замены ($|X| = 376$)

X	B194BAC8 0A08F53B 366D008E 584A5DE4 8504FA9D 1BB6C7AC 252E72C2 02FDCE0D 5BE3D612 17B96181 FE6786AD 716B89$_{16}$
θ	E9DEE72C 8F0C0FA6 2DDB49F4 6F739647 06075316 ED247A37 39CBA383 03A98BF6$_{16}$
Y	69CCA1C9 3557C9E3 D66BC3E0 FA88FA6E 36F00CFE D6D1CA14 98C12798 F4BEB207 5F23102E F1097107 75017F73 806DA9$_{16}$

Таблица А.8 — Расшифрование в режиме простой замены ($|X| = 384$)

X	E12BDC1A E28257EC 703FCCF0 95EE8DF1 C1AB7638 9FE678CA F7C6F860 D5BB9C4F F33C657B 637C306A DD4EA779 9EB23D31$_{16}$
θ	92BD9B1C E5D14101 5445FBC9 5E4D0EF2 682080AA 227D642F 2687F934 90405511$_{16}$
Y	0DC53006 00CAB840 B38448E5 E993F421 E55A239F 2AB5C5D5 FDB6E81B 40938E2A 54120CA3 E6E19C7A D750FC35 31DAEAB7$_{16}$

Таблица А.9 — Расшифрование в режиме простой замены ($|X| = 288$)

X	E12BDC1A E28257EC 703FCCF0 95EE8DF1 C1AB7638 9FE678CA F7C6F860 D5BB9C4F F33C657B$_{16}$
θ	92BD9B1C E5D14101 5445FBC9 5E4D0EF2 682080AA 227D642F 2687F934 90405511$_{16}$
Y	0DC53006 00CAB840 B38448E5 E993F421 5780A6E2 B69EAFBB 258726D7 B6718523 E55A239F$_{16}$

А.3 Шифрование в режиме сцепления блоков

В таблицах А.10, А.11 представлены примеры зашифрования в режиме сцепления блоков. В таблицах А.12, А.13 представлены примеры расшифрования в этом же режиме.

Таблица А.10 — Зашифрование в режиме сцепления блоков ($|X| = 384$)

X	B194BAC8 0A08F53B 366D008E 584A5DE4 8504FA9D 1BB6C7AC 252E72C2 02FDCE0D 5BE3D612 17B96181 FE6786AD 716B890B$_{16}$
θ	E9DEE72C 8F0C0FA6 2DDB49F4 6F739647 06075316 ED247A37 39CBA383 03A98BF6$_{16}$
S	BE329713 43FC9A48 A02A885F 194B09A1$_{16}$
Y	10116EFA E6AD58EE 14852E11 DA1B8A74 5CF2480E 8D03F1C1 9492E53E D3A70F60 657C1EE8 C0E0AE5B 58388BF8 A68E3309$_{16}$

Таблица А.11 — Зашифрование в режиме сцепления блоков ($|X| = 288$)

X	B194BAC8 0A08F53B 366D008E 584A5DE4 8504FA9D 1BB6C7AC 252E72C2 02FDCE0D 5BE3D612$_{16}$
θ	E9DEE72C 8F0C0FA6 2DDB49F4 6F739647 06075316 ED247A37 39CBA383 03A98BF6$_{16}$
S	BE329713 43FC9A48 A02A885F 194B09A1$_{16}$
Y	10116EFA E6AD58EE 14852E11 DA1B8A74 6A9BBADC AF73F968 F875DEDC 0A44F6B1 5CF2480E$_{16}$

Таблица А.12 — Расшифрование в режиме сцепления блоков ($|X| = 384$)

X	E12BDC1A E28257EC 703FCCF0 95EE8DF1 C1AB7638 9FE678CA F7C6F860 D5BB9C4F F33C657B 637C306A DD4EA779 9EB23D31$_{16}$
θ	92BD9B1C E5D14101 5445FBC9 5E4D0EF2 682080AA 227D642F 2687F934 90405511$_{16}$
S	7ECDA4D0 1544AF8C A58450BF 66D2E88A$_{16}$
Y	730894D6 158E17CC 1600185A 8F411CAB 0471FF85 C8379239 8D8924EB D57D03DB 95B97A9B 7907E4B0 20960455 E46176F8$_{16}$

Таблица А.13 — Расшифрование в режиме сцепления блоков ($|X| = 288$)

X	E12BDC1A E28257EC 703FCCF0 95EE8DF1 C1AB7638 9FE678CA F7C6F860 D5BB9C4F F33C657B$_{16}$
θ	92BD9B1C E5D14101 5445FBC9 5E4D0EF2 682080AA 227D642F 2687F934 90405511$_{16}$
S	7ECDA4D0 1544AF8C A58450BF 66D2E88A$_{16}$
Y	730894D6 158E17CC 1600185A 8F411CAB B6AB7AF8 541CF857 55B8EA27 239F08D2 166646E4$_{16}$

А.4 Шифрование в режиме гаммирования с обратной связью

В таблицах А.14, А.15 представлены примеры зашифрования и расшифрования в режиме гаммирования с обратной связью.

Таблица А.14 — Зашифрование в режиме гаммирования с обратной связью

X	B194BAC8 0A08F53B 366D008E 584A5DE4 8504FA9D 1BB6C7AC 252E72C2 02FDCE0D 5BE3D612 17B96181 FE6786AD 716B890B$_{16}$
θ	E9DEE72C 8F0C0FA6 2DDB49F4 6F739647 06075316 ED247A37 39CBA383 03A98BF6$_{16}$
S	BE329713 43FC9A48 A02A885F 194B09A1$_{16}$
Y	C31E490A 90EFA374 626CC99E 4B7B8540 A6E48685 464A5A06 849C9CA7 69A1B0AE 55C2CC59 39303EC8 32DD2FE1 6C8E5A1B$_{16}$

Таблица А.15 — Расшифрование в режиме гаммирования с обратной связью

X	E12BDC1A E28257EC 703FCCF0 95EE8DF1 C1AB7638 9FE678CA F7C6F860 D5BB9C4F F33C657B 637C306A DD4EA779 9EB23D31$_{16}$
θ	92BD9B1C E5D14101 5445FBC9 5E4D0EF2 682080AA 227D642F 2687F934 90405511$_{16}$
S	7ECDA4D0 1544AF8C A58450BF 66D2E88A$_{16}$
Y	FA9D107A 86F375EE 65CD1DB8 81224BD0 16AFF814 938ED39B 3361ABB0 BF0851B6 52244EB0 6842DD4C 94AA4500 774E40BB$_{16}$

А.5 Шифрование в режиме счетчика

В таблице А.16 представлен пример шифрования в режиме счетчика.

Таблица А.16 — Шифрование в режиме счетчика

X	B194BAC8 0A08F53B 366D008E 584A5DE4 8504FA9D 1BB6C7AC 252E72C2 02FDCE0D 5BE3D612 17B96181 FE6786AD 716B890B$_{16}$
θ	E9DEE72C 8F0C0FA6 2DDB49F4 6F739647 06075316 ED247A37 39CBA383 03A98BF6$_{16}$
S	BE329713 43FC9A48 A02A885F 194B09A1$_{16}$
Y	52C9AF96 FF50F644 35FC43DE F56BD797 D5B5B1FF 79FB4125 7AB9CDF6 E63E81F8 F0034147 3EAE4098 33622DE0 5213773A$_{16}$

А.6 Выработка имитовставки

В таблицах А.17, А.18 представлены примеры выработки имитовставки.

Таблица А.17 — Выработка имитовставки ($|X| = 104$)

X	B194BAC8 0A08F53B 366D008E 58$_{16}$
θ	E9DEE72C 8F0C0FA6 2DDB49F4 6F739647 06075316 ED247A37 39CBA383 03A98BF6$_{16}$
Y	7260DA60 138F96C9$_{16}$

Таблица А.18 — Выработка имитовставки ($|X| = 384$)

X	B194BAC8 0A08F53B 366D008E 584A5DE4 8504FA9D 1BB6C7AC 252E72C2 02FDCE0D 5BE3D612 17B96181 FE6786AD 716B890B$_{16}$
θ	E9DEE72C 8F0C0FA6 2DDB49F4 6F739647 06075316 ED247A37 39CBA383 03A98BF6$_{16}$
Y	2DAB5977 1B4B16D0$_{16}$

А.7 Шифрование и имитозащита данных

В таблице А.19 представлены примеры применения операции $*$. В таблицах А.20 и А.21 представлены примеры установки и снятия защиты данных.

Таблица А.19 — Операция $*$

u	34904055 11BE3297 1343724C 5AB793E9$_{16}$
v	22481783 8761A9D6 E3EC9689 110FB0F3$_{16}$
$u * v$	0001D107 FC67DE40 04DC2C80 3DFD95C3$_{16}$
u	703FCCF0 95EE8DF1 C1ABF8EE 8DF1C1AB$_{16}$
v	2055704E 2EDB48FE 87E74075 A5E77EB1$_{16}$
$u * v$	4A5C9593 8B3FE8F6 74D59BC1 EB356079$_{16}$

Таблица А.20 — Установка защиты данных

X	B194BAC8 0A08F53B 366D008E 584A5DE4$_{16}$
I	8504FA9D 1BB6C7AC 252E72C2 02FDCE0D 5BE3D612 17B96181 FE6786AD 716B890B$_{16}$
θ	E9DEE72C 8F0C0FA6 2DDB49F4 6F739647 06075316 ED247A37 39CBA383 03A98BF6$_{16}$
S	BE329713 43FC9A48 A02A885F 194B09A1$_{16}$
Y	52C9AF96 FF50F644 35FC43DE F56BD797$_{16}$
T	3B2E0AEB 2B91854B$_{16}$

Таблица А.21 — Снятие защиты данных

X	E12BDC1A E28257EC 703FCCF0 95EE8DF1$_{16}$
I	C1AB7638 9FE678CA F7C6F860 D5BB9C4F F33C657B 637C306A DD4EA779 9EB23D31$_{16}$
T	6A2C2C94 C4150DC0$_{16}$
θ	92BD9B1C E5D14101 5445FBC9 5E4D0EF2 682080AA 227D642F 2687F934 90405511$_{16}$
S	7ECDA4D0 1544AF8C A58450BF 66D2E88A$_{16}$
Y	DF181ED0 08A20F43 DCBBB936 50DAD34B$_{16}$

А.8 Шифрование и имитозащита ключа

В таблицах А.22 и А.23 представлены примеры установки и снятия защиты ключа.

Таблица А.22 — Установка защиты ключа

X	B194BAC8 0A08F53B 366D008E 584A5DE4 8504FA9D 1BB6C7AC 252E72C2 02FDCE0D$_{16}$
I	5BE3D612 17B96181 FE6786AD 716B890B$_{16}$
θ	E9DEE72C 8F0C0FA6 2DDB49F4 6F739647 06075316 ED247A37 39CBA383 03A98BF6$_{16}$
Y	49A38EE1 08D6C742 E52B774F 00A6EF98 B106CBD1 3EA4FB06 80323051 BC04DF76 E487B055 C69BCF54 1176169F 1DC9F6C8$_{16}$

Таблица А.23 — Снятие защиты ключа

X	E12BDC1A E28257EC 703FCCF0 95EE8DF1 C1AB7638 9FE678CA F7C6F860 D5BB9C4F F33C657B 637C306A DD4EA779 9EB23D31$_{16}$
I	B5EF68D8 E4A39E56 7153DE13 D72254EE$_{16}$
θ	92BD9B1C E5D14101 5445FBC9 5E4D0EF2 682080AA 227D642F 2687F934 90405511$_{16}$
Y	92632EE0 C21AD9E0 9A39343E 5C07DAA4 889B03F2 E6847EB1 52EC99F7 A4D9F154$_{16}$

А.9 Хэширование

В таблицах А.24, А.25 и А.26 представлены примеры хэширования.

Таблица А.24 — Хэширование ($|X| = 104$)

X	B194BAC8 0A08F53B 366D008E 58_{16}
Y	ABEF9725 D4C5A835 97A367D1 4494CC25 42F20F65 9DDFECC9 61A3EC55 0CBA8C75$_{16}$

Таблица А.25 — Хэширование ($|X| = 256$)

X	B194BAC8 0A08F53B 366D008E 584A5DE4 8504FA9D 1BB6C7AC 252E72C2 02FDCE0D$_{16}$
Y	749E4C36 53AECE5E 48DB4761 227742EB 6DBE13F4 A80F7BEF F1A9CF8D 10EE7786$_{16}$

Таблица А.26 — Хэширование ($|X| = 384$)

X	B194BAC8 0A08F53B 366D008E 584A5DE4 8504FA9D 1BB6C7AC 252E72C2 02FDCE0D 5BE3D612 17B96181 FE6786AD 716B890B$_{16}$
Y	9D02EE44 6FB6A29F E5C982D4 B13AF9D3 E90861BC 4CEF27CF 306BFB0B 174A154A$_{16}$

А.10 Расширение ключа

В таблицах А.27, А.28 представлены примеры расширения ключа.

Таблица А.27 — Расширение ключа ($n = 4$)

θ_1	E9DEE72C$_{16}$
θ_2	8F0C0FA6$_{16}$
θ_3	2DDB49F4$_{16}$
θ_4	6F739647$_{16}$
θ	E9DEE72C 8F0C0FA6 2DDB49F4 6F739647 E9DEE72C 8F0C0FA6 2DDB49F4 6F739647$_{16}$

Таблица А.28 — Расширение ключа ($n = 6$)

θ_1	E9DEE72C$_{16}$
θ_2	8F0C0FA6$_{16}$
θ_3	2DDB49F4$_{16}$
θ_4	6F739647$_{16}$
θ_5	06075316$_{16}$
θ_6	ED247A37$_{16}$
θ	E9DEE72C 8F0C0FA6 2DDB49F4 6F739647 06075316 ED247A37 4B09A17E 8450BF66$_{16}$

А.11 Преобразование ключа

В таблицах А.29, А.30, А.31 представлены примеры преобразования ключа.

Таблица А.29 — Преобразование ключа ($m = 128$)

X	E9DEE72C 8F0C0FA6 2DDB49F4 6F739647 06075316 ED247A37 39CBA383 03A98BF6$_{16}$
D	01000000 00000000 00000000$_{16}$
I	5BE3D612 17B96181 FE6786AD 716B890B$_{16}$
m	128
Y	6BBBC233 6670D31A B83DAA90 D52C0541$_{16}$

Таблица А.30 — Преобразование ключа ($m = 192$)

X	E9DEE72C 8F0C0FA6 2DDB49F4 6F739647 06075316 ED247A37 39CBA383 03A98BF6$_{16}$
D	01000000 00000000 00000000$_{16}$
I	5BE3D612 17B96181 FE6786AD 716B890B$_{16}$
m	192
Y	9A2532A1 8CBAF145 398D5A95 FEEA6C82 5B9C1971 56A00275$_{16}$

Таблица А.31 — Преобразование ключа ($m = 256$)

X	E9DEE72C 8F0C0FA6 2DDB49F4 6F739647 06075316 ED247A37 39CBA383 03A98BF6$_{16}$
D	01000000 00000000 00000000$_{16}$
I	5BE3D612 17B96181 FE6786AD 716B890B$_{16}$
m	256
Y	76E166E6 AB21256B 6739397B 672B8796 14B81CF0 5955FC3A B09343A7 45C48F77$_{16}$

Приложение Б
(рекомендуемое)
Модуль ACH.1

В модуле ACH.1 идентификаторы алгоритмов шифрования и имитозащиты снабжаются трехзначным кодом `NNN`, который обозначает длину используемого ключа: `NNN` = 128, 192 или 256.

Алгоритмы стандарта обозначаются следующим образом:

`belt-ecbNNN`	алгоритмы шифрования в режиме простой замены (6.2);
`belt-cbcNNN`	алгоритмы шифрования в режиме сцепления блоков (6.3);
`belt-cfbNNN`	алгоритмы шифрования в режиме гаммирования с обратной связью (6.4);
`belt-ctrNNN`	алгоритмы шифрования в режиме счетчика (6.5);
`belt-macNNN`	алгоритм выработки имитовставки (6.6);
`belt-datawrapNNN`	алгоритмы одновременного шифрования и имитозащиты данных (6.7);
`belt-keywrapNNN`	алгоритмы одновременного шифрования и имитозащиты ключа (6.8);
`belt-hash256`	алгоритм хэширования (6.9);
`belt-keyexpand`	алгоритм расширения ключа (7.1);
`belt-keyrep`	алгоритм преобразования ключа (7.2).

В модуле ACH.1 определяются форматы следующих параметров:

`IV`	синхропосылка в алгоритмах `belt-cbcNNN`, `belt-cfbNNN`, `belt-ctrNNN`, `belt-datawrapNNN`;
`KeyHeader`	заголовок ключа в алгоритмах `belt-keywrapNNN`, `belt-keyrep`;
`KeyLevel`	уровень ключа в алгоритме `belt-keyrep`.

Модуль ACH.1 имеет следующий вид:

```
Belt-module-v1 {iso(1) member-body(2) by(112) 0 2 0 34 101 31 module(1) ver1(1)}
DEFINITIONS ::=
BEGIN
  belt OBJECT IDENTIFIER ::= {iso(1) member-body(2) by(112) 0 2 0 34 101 31}

  belt-ecb128 OBJECT IDENTIFIER ::= {belt 11}
  belt-ecb192 OBJECT IDENTIFIER ::= {belt 12}
  belt-ecb256 OBJECT IDENTIFIER ::= {belt 13}
  belt-cbc128 OBJECT IDENTIFIER ::= {belt 21}
  belt-cbc192 OBJECT IDENTIFIER ::= {belt 22}
  belt-cbc256 OBJECT IDENTIFIER ::= {belt 23}
```

```
  belt-cfb128 OBJECT IDENTIFIER ::= {belt 31}
  belt-cfb192 OBJECT IDENTIFIER ::= {belt 32}
  belt-cfb256 OBJECT IDENTIFIER ::= {belt 33}
  belt-ctr128 OBJECT IDENTIFIER ::= {belt 41}
  belt-ctr192 OBJECT IDENTIFIER ::= {belt 42}
  belt-ctr256 OBJECT IDENTIFIER ::= {belt 43}
  belt-mac128 OBJECT IDENTIFIER ::= {belt 51}
  belt-mac192 OBJECT IDENTIFIER ::= {belt 52}
  belt-mac256 OBJECT IDENTIFIER ::= {belt 53}
  belt-datawrap128 OBJECT IDENTIFIER ::= {belt 61}
  belt-datawrap192 OBJECT IDENTIFIER ::= {belt 62}
  belt-datawrap256 OBJECT IDENTIFIER ::= {belt 63}
  belt-keywrap128 OBJECT IDENTIFIER ::= {belt 71}
  belt-keywrap192 OBJECT IDENTIFIER ::= {belt 72}
  belt-keywrap256 OBJECT IDENTIFIER ::= {belt 73}
  belt-hash256 OBJECT IDENTIFIER ::= {belt 81}
  belt-keyexpand OBJECT IDENTIFIER ::= {belt 91}
  belt-keyrep OBJECT IDENTIFIER ::= {belt 101}

  IV ::= OCTET STRING (SIZE(16))
  KeyHeader ::= OCTET STRING (SIZE(16))
  KeyLevel ::= OCTET STRING (SIZE(12))
END
```

Библиография

[1] Лидл Р., Нидеррайтер Г. Конечные поля
М.: Мир, 1988

Поправка к официальной редакции

В каком месте	Напечатано	Должно быть
Пункт 6.1.3, шаг 1	$c \leftarrow X_4$	$d \leftarrow X_4$
Пункт 6.8.2	$r_1 = r^{**} \parallel r^*$	$r = r^{**} \parallel r^*$
Приложение Б	В модуль АСН.1 определяются форматы следующих параметров:	В модуле АСН.1 определяются форматы следующих параметров: