ПРЕДВАРИТЕЛЬНЫЙ ГОСУДАРСТВЕННЫЙ СТАНДАРТ РЕСПУБЛИКИ БЕЛАРУСЬ

СТБ П 34.101.31–2007

Информационные технологии
Защита информации
КРИПТОГРАФИЧЕСКИЕ АЛГОРИТМЫ ШИФРОВАНИЯ И КОНТРОЛЯ ЦЕЛОСТНОСТИ

Інфармацыйныя тэхналогіі
Ахова інфармацыі
КРЫПТАГРАФІЧНЫЯ АЛГАРЫТМЫ ШЫФРАВАННЯ І КАНТРОЛЮ ЦЭЛАСНАСЦІ

Настоящий проект предстандарта не подлежит применению до его утверждения

Госстандарт
Минск

УДК МКС 35.240.40 КП 05

Ключевые слова: технологии информационные, шифрование, имитозащита, хэширование

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 «Национальный научно–исследовательский центр прикладных проблем математики и информатики»

ВНЕСЕН Государственным центром безопасности информации при Президенте Республики Беларусь (ГЦБИ)

2 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ постановлением Госстандарта Республики Беларусь от 200 г. № в качестве предварительного государственного стандарта Республики Беларусь со сроком действия с . .200 г. по . .200 г.

3 ВВЕДЕН ВПЕРВЫЕ

4 Срок представления разработчику предстандарта замечаний и предложений, в том числе о целесообразности (нецелесообразности) перевода предстандарта в государственный стандарт, до . .200 г.

Адрес: 220030, г. Минск, пр. Независимости, 4

Факс: (017) 2095104

Телефон разработчика: (017) 2095071

E-mail: nrcapmi@bsu.by

Издан на русском языке

Содержание

ПРЕДВАРИТЕЛЬНЫЙ 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
Data security
Data Encryption and Integrity Algorithms

Дата введения с 01.02.2008
Дата окончания действия 01.02.2010

1 Область применения

Настоящий предстандарт определяет семейство криптографических алгоритмов шифрования и контроля целостности, которые используются в криптографических методах защиты информации при хранении, передаче и обработке данных.

Настоящий предстандарт применяется при разработке средств криптографической защиты информации в автоматизированных системах.

2 Обозначения

$\{0,1\}^n$	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, $m \le \|u\|$;
$u \parallel v$	конкатенация $u_1u_2\dots u_nv_1v_2\dots v_m$ слов $u = u_1u_2\dots u_n$ и $v = v_1v_2\dots v_m$;
$\texttt{01234...}_{16}$	представление $u \in \{0,1\}^{4*}$ шестнадцатеричным словом, при котором последовательным четырем символам u соответствует один шестнадцатеричный символ (например, $10100010 = \texttt{A2}_{16}$);
$u \oplus v$	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 $w_i \equiv u_i + v_i \pmod 2$;

Проект, окончательная редакция

$\bar{u}$	а) для $u = u_1 u_2 \dots u_8 \in \{0,1\}^8$ число $2^7 u_1 + 2^6 u_2 + \dots + u_8$ 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 числа U слово $u \in \{0,1\}^{8n}$ такое, что $\bar{u} \equiv U \pmod{2^{8n}}$;
$u \boxplus v$	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 $u = v \boxplus w$;
$\lambda(u)$	для $u \in \{0,1\}^{32}$ слово $\langle 2\bar{u} \rangle_{32}$, если $\bar{u} < 2^{31}$, или слово $\langle 2\bar{u} + 1 \rangle_{32}$, если $\bar{u} \geq 2^{31}$;
$\lambda^r(u)$	слово, полученное r-кратным действием λ на u;
$a \leftarrow u$	присвоение переменной a значения u;
$a \leftrightarrow b$	перестановка значений переменных a и b;
$F_\theta(X)$	результат зашифрования слова $X \in \{0,1\}^{128}$ на ключе $\theta \in \{0,1\}^{256}$ по алгоритму из 4.1.3;
$F_\theta^{-1}(X)$	результат расшифрования слова $X \in \{0,1\}^{128}$ на ключе $\theta \in \{0,1\}^{256}$ по алгоритму из 4.1.4.

3 Общие положения

3.1 Назначение

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

Криптографические алгоритмы предстандарта построены на основе вспомогательных алгоритмов шифрования блоков данных — двоичных слов длины 128. Криптографические алгоритмы делятся на шесть групп:

1) алгоритмы шифрования в режиме простой замены;
2) алгоритмы шифрования в режиме сцепления блоков;
3) алгоритмы шифрования в режиме гаммирования с обратной связью;
4) алгоритмы шифрования в режиме счетчика;
5) алгоритм выработки имитовставки;
6) алгоритм хэширования.

Первые четыре группы предназначены для обеспечения конфиденциальности. Каждая группа включает алгоритм зашифрования и алгоритм расшифрования на секретном ключе. Алгоритм зашифрования преобразует открытые данные в защищенные, а алгоритм расшифрования выполняет обратное преобразование. Стороны, располагающие общим секретным ключом, могут организовать защищенный обмен данными путем их зашифрования перед отправкой и расшифрования после получения.

Пятый алгоритм предназначен для создания имитовставок — контрольных данных, которые определяются с использованием ключа. Стороны, располагающие общим секретным ключом, могут организовать контроль целостности при передаче данных путем добавления к ним имитовставок при отправке и проверки имитовставок при получении.

Проверка имитовставок дополнительно позволяет стороне-получателю убедиться в знании стороной-отправителем секретного ключа.

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

3.2 Ключ

В алгоритмах шифрования и выработки имитовставки используется ключ $\theta \in \{0,1\}^{256}$, который однозначно определяет криптографическое преобразование данных.

Ключ должен вырабатываться без возможности предсказания, распространяться с соблюдением мер конфиденциальности и храниться в секрете. Один и тот же ключ не должен использоваться в алгоритмах различных групп.

Разрешается использовать ключ θ длины 256, который является результатом расширения короткого ключа длины 128 или 192. Пусть короткий ключ имеет вид $\theta_1 \parallel \theta_2 \parallel \ldots \parallel \theta_d$, где $\theta_i \in \{0,1\}^{32}$ и $d = 4$ или $d = 6$. Процедура расширения состоят в определении слов $\theta_{d+1}, \ldots, \theta_8 \in \{0,1\}^{32}$ с последующим формированием $\theta = \theta_1 \parallel \theta_2 \parallel \ldots \parallel \theta_8$. Должны использоваться следующие правила расширения:

1) при $d = 4$ установить: $\theta_5 \leftarrow \theta_1$, $\theta_6 \leftarrow \theta_2$, $\theta_7 \leftarrow \theta_3$, $\theta_8 \leftarrow \theta_4$;
2) при $d = 6$ установить: $\theta_7 \leftarrow \theta_1 \oplus \theta_2 \oplus \theta_3$, $\theta_8 \leftarrow \theta_4 \oplus \theta_5 \oplus \theta_6$.

3.3 Синхропосылка

При шифровании в режимах сцепления блоков, гаммирования с обратной связью и счетчика используется синхропосылка $S \in \{0,1\}^{128}$, которая обеспечивает уникальность криптографических преобразований на фиксированном ключе.

Синхропосылка не является секретным параметром, может добавляться к зашифрованным данным и передаваться вместе с ними. При зашифровании на одном и том же ключе должны использоваться различные синхропосылки.

3.4 Имитовставка

Имитовставкой слова $X \in \{0,1\}^*$ на ключе $\theta \in \{0,1\}^{256}$ является слово $Y \in \{0,1\}^{64}$. Разрешается использовать не все 64 символа имитовставки, а только первые $l \leq 64$ символов. При выборе l следует учитывать, что вероятность угадывания имитовставки заданного слова без знания ключа равняется 2^{-l}.

3.5 Хэш-значение

Хэш-значением слова $X \in \{0,1\}^*$ является слово $Y \in \{0,1\}^{256}$. Разрешается использовать не все 256 символов хэш-значения, а только первые $l \leq 256$ символов. При выборе l следует учитывать, что для определения слова с заданным хэш-значением требуется вы-

полнить порядка 2^l операций, а для определения двух различных слов с одинаковыми хэш-значениями требуется выполнить порядка $2^{l/2}$ операций.

4 Алгоритмы шифрования и контроля целостности

4.1 Шифрование блоков данных

4.1.1 Входные и выходные данные

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

Выходными данными является слово $Y \in \{0,1\}^{128}$, 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 $K_1 = \theta_1, K_2 = \theta_2, \ldots, K_8 = \theta_8, K_9 = \theta_1, K_{10} = \theta_2, \ldots, K_{56} = \theta_8$.

4.1.2 Переменные и вспомогательные преобразования

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

Подстановка H. Подстановка $H\colon \{0,1\}^8 \to \{0,1\}^8$ задается таблицей 1. В таблице 1 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 1 — 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

$$G_r(u) = \lambda^r\left(H(u_1) \parallel H(u_2) \parallel H(u_3) \parallel H(u_4)\right).$$

4.1.3 Алгоритм зашифрования

Для зашифрования слова 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. рис. 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.

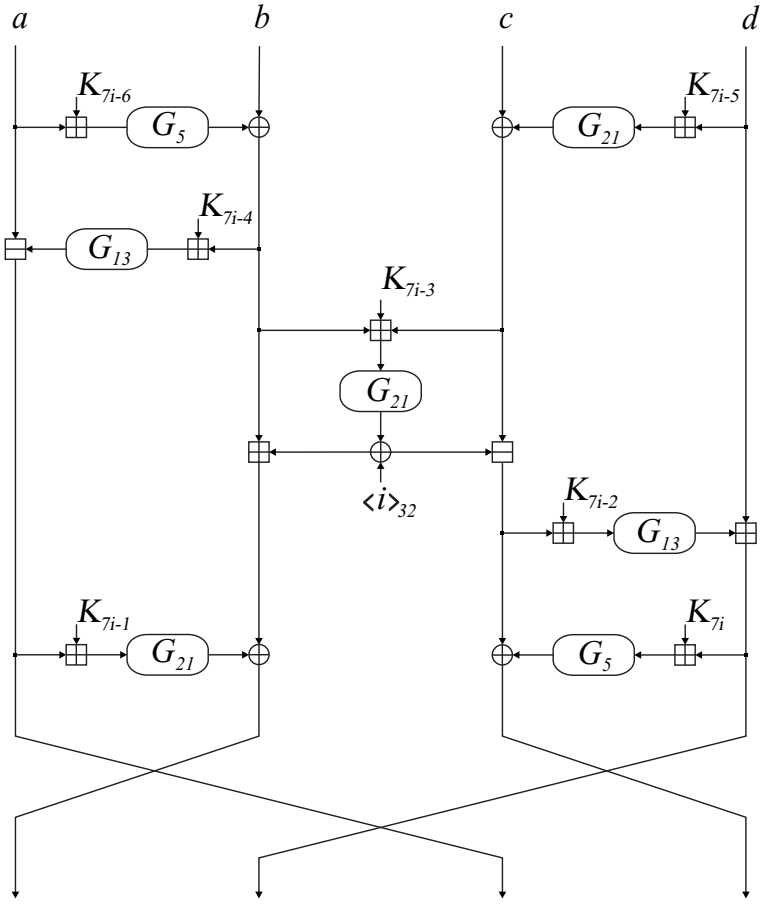

Рисунок 1 — Вычисления на i-м такте зашифрования

4.1.4 Алгоритм расшифрования

Для расшифрования слова 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.

4.2 Шифрование в режиме простой замены

4.2.1 Входные и выходные данные

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

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

Входное слово X записывается в виде

$$X = X_1 \parallel X_2 \parallel \ldots \parallel X_d, \quad X_i \in \{0,1\}^{128}.$$

При шифровании словам X_i ставятся в соответствие слова $Y_i \in \{0,1\}^{128}$, из которых затем составляется Y.

4.2.2 Алгоритм зашифрования

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

1 Для $i = 1, 2, \ldots, d$ выполнить $Y_i \leftarrow F_\theta(X_i)$.

2 Установить $Y \leftarrow Y_1 \parallel Y_2 \parallel \ldots \parallel Y_d$.

3 Возвратить Y.

4.2.3 Алгоритм расшифрования

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

1 Для $i = 1, 2, \ldots, d$ выполнить $Y_i \leftarrow F_\theta^{-1}(X_i)$.

2 Установить $Y \leftarrow Y_1 \parallel Y_2 \parallel \ldots \parallel Y_d$.

3 Возвратить Y.

4.3 Шифрование в режиме сцепления блоков

4.3.1 Входные и выходные данные

Входными данными алгоритмов зашифрования и расшифрования являются слово $X \in \{0,1\}^{128*}$, 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 слово X записывается в виде

$$X = X_1 \parallel X_2 \parallel \ldots \parallel X_d, \quad X_i \in \{0,1\}^{128}.$$

При шифровании словам X_i ставятся в соответствие слова $Y_i \in \{0,1\}^{128}$, из которых затем составляется Y.

4.3.2 Алгоритм зашифрования

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

1 Для $i = 1, 2, \ldots, d$ выполнить $Y_i \leftarrow F_\theta(X_i \oplus Y_{i-1})$, где $Y_0 = F_\theta(S)$.

2 Установить $Y \leftarrow Y_1 \parallel Y_2 \parallel \ldots \parallel Y_d$.

3 Возвратить Y.

4.3.3 Алгоритм расшифрования

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

1 Для $i = 1, 2, \ldots, d$ выполнить $Y_i \leftarrow F_\theta^{-1}(X_i) \oplus X_{i-1}$, где $X_0 = F_\theta(S)$.

2 Установить $Y \leftarrow Y_1 \parallel Y_2 \parallel \ldots \parallel Y_d$.

3 Возвратить Y.

4.4 Шифрование в режиме гаммирования с обратной связью

4.4.1 Входные и выходные данные

Входными данными алгоритмов зашифрования и расшифрования являются слово $X \in \{0,1\}^*$, 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 слово X записывается в виде

$$X = X_1 \parallel X_2 \parallel \ldots \parallel X_d, \quad |X_1| = |X_2| = \ldots = |X_{d-1}| = 128, \quad |X_d| \le 128.$$

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

4.4.2 Алгоритм зашифрования

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

1 Для $i = 1, 2, \ldots, d$ выполнить $Y_i \leftarrow X_i \oplus L_{|X_i|}(F_\theta(Y_{i-1}))$, где $Y_0 = S$.

2 Установить $Y \leftarrow Y_1 \parallel Y_2 \parallel \ldots \parallel Y_d$.

3 Возвратить Y.

4.4.3 Алгоритм расшифрования

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

1 Для $i = 1, 2, \ldots, d$ выполнить $Y_i \leftarrow X_i \oplus L_{|X_i|}(F_\theta(X_{i-1}))$, где $X_0 = S$.

2 Установить $Y \leftarrow Y_1 \parallel Y_2 \parallel \ldots \parallel Y_d$.

3 Возвратить Y.

4.5 Шифрование в режиме счетчика

4.5.1 Входные и выходные данные

Входными данными алгоритмов зашифрования и расшифрования являются слово $X \in \{0,1\}^*$, 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 слово X записывается в виде

$$X = X_1 \parallel X_2 \parallel \ldots \parallel X_d, \quad |X_1| = |X_2| = \ldots = |X_{d-1}| = 128, \quad |X_d| \le 128.$$

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

4.5.2 Переменные

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

4.5.3 Алгоритм зашифрования

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

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

2 Для $i = 1, 2, \ldots, d$ 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 $Y \leftarrow Y_1 \parallel Y_2 \parallel \ldots \parallel Y_d$.

4 Возвратить Y.

4.5.4 Алгоритм расшифрования

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

4.6 Выработка имитовставки

4.6.1 Входные и выходные данные

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

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

Входное слово X записывается в виде

$$X = X_1 \parallel X_2 \parallel \ldots \parallel X_d, \quad |X_1| = |X_2| = \ldots = |X_{d-1}| = 128, \quad |X_d| \le 128,$$

где $|X_d| = 0$ только если X — пустое слово.

4.6.2 Переменные и вспомогательные преобразования

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

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.

4.6.3 Алгоритм

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

1 Установить $s \leftarrow 0^{128}$, $r \leftarrow F_\theta(s)$.
2 Для $i = 1, 2, \ldots, d-1$ выполнить $s \leftarrow F_\theta(s \oplus X_i)$.
3 Если $|X_d| = 128$, то $s \leftarrow s \oplus X_d \oplus \varphi_1(r)$, иначе $s \leftarrow s \oplus \psi(X_d) \oplus \varphi_2(r)$.
4 Установить $Y \leftarrow L_{64}(F_\theta(s))$.
5 Возвратить Y.

4.7 Хэширование

4.7.1 Входные и выходные данные

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

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

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

$$X \parallel 0^t = X_1 \parallel X_2 \parallel \ldots \parallel X_d, \quad X_i \in \{0,1\}^{256}.$$

4.7.2 Переменные и вспомогательные преобразования

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

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

4.7.3 Алгоритм

Хэширование слова X состоит в выполнении следующих шагов:

1 Установить $s \leftarrow 0^{128}$.

2 Установить

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

где присваиваемое значение определяется последовательными (слева направо и сверху вниз) элементами первых двух строк таблицы 1.

3 Для $i = 1, 2, \ldots, d$ 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.