

---

ФЕДЕРАЛЬНОЕ АГЕНТСТВО  
ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

---



НАЦИОНАЛЬНЫЙ  
СТАНДАРТ  
РОССИЙСКОЙ  
ФЕДЕРАЦИИ

ГОСТ Р  
34.13—  
2015

---

Информационная технология

**КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА  
ИНФОРМАЦИИ**

Режимы работы блочных шифров

Издание официальное



Москва  
Стандартинформ  
2015

## Предисловие

1 РАЗРАБОТАН Центром защиты информации и специальной связи ФСБ России с участием Открытого акционерного общества «Информационные технологии и коммуникационные системы» (ОАО «ИнфоТеКС»)

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 26 «Криптографическая защита информации»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 19 июня 2015 г. № 750-ст

4 ВЗАМЕН ГОСТ Р ИСО/МЭК 10116–93

*Правила применения настоящего стандарта установлены в ГОСТ Р 1.0–2012 (раздел 8). Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок – в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования – на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет ([www.gost.ru](http://www.gost.ru))*

© Стандартиформ, 2015

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

## Содержание

1	Область применения .....
2	Термины, определения и обозначения .....
2.1	Термины и определения .....
2.2	Обозначения .....
3	Общие положения .....
4	Вспомогательные операции .....
4.1	Дополнение сообщения .....
4.1.1	Процедура 1 .....
4.1.2	Процедура 2 .....
4.1.3	Процедура 3 .....
4.2	Выработка начального значения .....
4.3	Процедура усечения .....
5	Режимы работы алгоритмов блочного шифрования .....
5.1	Режим простой замены .....
5.2	Режим гаммирования .....
5.3	Режим гаммирования с обратной связью по выходу .....
5.4	Режим простой замены с зацеплением .....
5.5	Режим гаммирования с обратной связью по шифртексту .....
5.6	Режим выработки имитовставки .....
	Приложение А (справочное) Контрольные примеры .....
	Библиография .....

## Введение

Настоящий стандарт содержит описание режимов работы блочных шифров. Данные режимы работы блочных шифров определяют правила криптографического преобразования данных и выработки имитовставки для сообщений произвольного размера.

Стандарт разработан взамен ГОСТ Р ИСО/МЭК 10116–93 «Информационная технология. Режимы работы для алгоритма  $n$ -разрядного блочного шифрования». Необходимость разработки настоящего стандарта вызвана потребностью в определении режимов работы блочных шифров, соответствующих современным требованиям к криптографической стойкости.

Настоящий стандарт терминологически и концептуально увязан с международными стандартами ИСО/МЭК 9797-1 [1], ИСО/МЭК 10116 [2], ИСО/МЭК 10118-1 [3], ИСО/МЭК 18033 [4], ИСО/МЭК 14888-1 [5].

П р и м е ч а н и е – Основная часть стандарта дополнена приложением А.

# НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

---

## Информационная технология

### КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ

#### Режимы работы блочных шифров

Information technology. Cryptographic data security.

Modes of operation for block ciphers

---

Дата введения — 2016—01—01

## 1 Область применения

Режимы работы блочных шифров, определенные в настоящем стандарте, рекомендуется использовать при разработке, производстве, эксплуатации и модернизации средств криптографической защиты информации в системах обработки информации различного назначения.

Настоящим стандартом следует руководствоваться, если информация конфиденциального характера подлежит защите в соответствии с законодательством Российской Федерации.

## 2 Термины, определения и обозначения

### 2.1 Термины и определения

В настоящем стандарте применены следующие термины с соответствующими определениями.

#### 2.1.1

<p><b>алгоритм зашифрования</b> (encryption algorithm): Алгоритм, реализующий зашифрование, т.е. преобразующий открытый текст в шифртекст. [ИСО/МЭК 18033–1, статья 2.19]</p>
---

### 2.1.2

**алгоритм расшифрования** (decryption algorithm): Алгоритм, реализующий расшифрование, т.е. преобразующий шифртекст в открытый текст.  
[ИСО/МЭК 18033–1, статья 2.14]

### 2.1.3

**базовый блочный шифр** (basic block cipher): Блочный шифр, реализующий при каждом фиксированном значении ключа одно обратимое отображение множества блоков открытого текста фиксированной длины в блоки шифртекста такой же длины.

### 2.1.4

**блок** (block): Строка бит определенной длины.  
[ИСО/МЭК 18033–1, статья 2.6]

### 2.1.5

**блочный шифр** (block cipher): Шифр из класса симметричных криптографических методов, в котором алгоритм зашифрования применяется к блокам открытого текста для получения блоков шифртекста.  
[ИСО/МЭК 18033–1, статья 2.7]

**Примечание** – В настоящем стандарте установлено, что термины «блочный шифр» и «алгоритм блочного шифрования» являются синонимами.

### 2.1.6

**дополнение** (padding): Приписывание дополнительных бит к строке бит.  
[ИСО/МЭК 10118–1, статья 3.9]

### 2.1.7

**зацепление блоков** (block chaining): Шифрование информации таким образом, что каждый блок шифртекста криптографически зависит от предыдущего блока шифртекста.  
[ИСО/МЭК 10116, статья 3.1]

### 2.1.8

**зашифрование** (encryption): Обратимое преобразование данных с помощью шифра, который формирует шифртекст из открытого текста.  
[ИСО/МЭК 18033–1, статья 2.18]

### 2.1.9

**имитовставка** (message authentication code): Строка бит фиксированной длины, полученная применением симметричного криптографического метода к сообщению, добавляемая к сообщению для обеспечения его целостности и аутентификации источника данных.  
[ИСО/МЭК 9797-1, статьи 3.9, 3.10]

## 2.1.10

**ключ** (key): Изменяемый параметр в виде последовательности символов, определяющий криптографическое преобразование.  
[ИСО/МЭК 18033–1, статья 2.21]

**Примечание** – В настоящем стандарте рассматриваются ключи только в виде последовательности двоичных символов (битов).

## 2.1.11

**начальное значение** (starting variable): Значение, возможно, полученное из синхропосылки и используемое для задания начальной точки режима работы блочного шифра.  
[ИСО/МЭК 10116, статья 3.12]

## 2.1.12

**открытый текст** (plaintext): Незашифрованная информация.  
[ИСО/МЭК 10116, статья 3.11]

## 2.1.13

**расшифрование** (decryption): Операция, обратная к зашифрованию.  
[ИСО/МЭК 18033-1, статья 2.13]

**Примечание** – В настоящем стандарте в целях сохранения терминологической преемственности по отношению к опубликованным научно-техническим изданиям применяется термин «шифрование», объединяющий операции, определенные терминами «зашифрование» и «расшифрование». Конкретное значение термина «шифрование» определяется в зависимости от контекста упоминания.

## 2.1.14

**симметричный криптографический метод** (symmetric cryptographic technique): Криптографический метод, использующий один и тот же ключ для преобразования, осуществляемого отправителем, и преобразования, осуществляемого получателем.  
[ИСО/МЭК 18033–1, статья 2.32]

## 2.1.15

**синхропосылка** (initializing value): Комбинация знаков, передаваемая по каналу связи и предназначенная для инициализации алгоритма шифрования.

## 2.1.16

**сообщение** (message): Строка бит произвольной конечной длины.  
[ИСО/МЭК 14888–1 статья 3.10]

## 2.1.17

**счетчик** (counter): Строка бит длины, равной длине блока блочного шифра, используемая при шифровании в режиме гаммирования.  
[ИСО/МЭК 10116, статья 3.4]

## 2.1.18

**шифр** (cipher): Криптографический метод, используемый для обеспечения конфиденциальности данных, включающий алгоритм зашифрования и алгоритм расшифрования.

[ИСО/МЭК 18033–1, статья 2.20]

## 2.1.19

**шифртекст** (ciphertext): Данные, полученные в результате зашифрования открытого текста с целью скрытия его содержания.

[ИСО/МЭК 10116, статья 3.3]

## 2.2 Обозначения

В настоящем стандарте используются следующие обозначения:

- $V^*$  – множество всех двоичных строк конечной длины, включая пустую строку;
- $V_s$  – множество всех двоичных строк длины  $s$ , где  $s$  – целое неотрицательное число; нумерация подстрок и компонент строки осуществляется справа налево начиная с нуля;
- $|A|$  – число компонент (длина) строки  $A \in V^*$  (если  $A$  – пустая строка, то  $|A| = 0$ );
- $A||B$  – конкатенация строк  $A, B \in V^*$ , т.е. строка из  $V_{|A|+|B|}$ , в которой подстрока с большими номерами компонент из  $V_{|A|}$  совпадает со строкой  $A$ , а подстрока с меньшими номерами компонент из  $V_{|B|}$  совпадает со строкой  $B$ ;
- $0^r$  – строка, состоящая из  $r$  нулей;
- $\oplus$  – операция покомпонентного сложения по модулю 2 двух двоичных строк одинаковой длины;
- $\mathbb{Z}_2^s$  – кольцо вычетов по модулю  $2^s$ ;
- $\boxplus_s$  – операция сложения в кольце  $\mathbb{Z}_2^s$ ;
- $x \bmod \ell$  – операция вычисления остатка от деления целого числа  $x$  на целое положительное число  $\ell$ ;



- $MSB_s: V^* \setminus \bigcup_{i=0}^{s-1} V_i \rightarrow V_s$
- отображение, ставящее в соответствие строке  $z_{m-1} \parallel \dots \parallel z_1 \parallel z_0$ ,  $m \geq s$ , строку  $z_{m-1} \parallel \dots \parallel z_{m-s+1} \parallel z_{m-s}$ ,  $z_i \in V_1, i = 0, 1, \dots, m-1$ ;
- $LSB_s: V^* \setminus \bigcup_{i=0}^{s-1} V_i \rightarrow V_s$
- отображение, ставящее в соответствие строке  $z_{m-1} \parallel \dots \parallel z_1 \parallel z_0$ ,  $m \geq s$ , строку  $z_{s-1} \parallel \dots \parallel z_1 \parallel z_0$ ,  $z_i \in V_1, i = 0, 1, \dots, m-1$ ;
- $A \ll r$
- операция логического сдвига строки  $A$  на  $r$  компонент в сторону компонент, имеющих большие номера. Если  $A \in V_s$ , то  $A \ll r \in V_s$ , причем
 
$$A \ll r = \begin{cases} LSB_{s-r}(A) \parallel 0^r, & \text{если } r < s, \\ 0^s, & \text{если } r \geq s; \end{cases}$$
- $Poly_s: V_s \rightarrow GF(2)[x]$
- отображение, ставящее в соответствие строке  $z = (z_{s-1} \parallel \dots \parallel z_0) \in V_s$  многочлен  $Poly_s(z) = \sum_{i=0}^{s-1} z_i x^i$ ;
- $Vec_s: \mathbb{Z}_2^s \rightarrow V_s$
- биективное отображение, сопоставляющее элементу кольца  $\mathbb{Z}_2^s$  его двоичное представление, т.е. для любого элемента  $z \in \mathbb{Z}_2^s$ , представленного в виде  $z = z_0 + 2 \cdot z_1 + \dots + 2^{s-1} \cdot z_{s-1}$ , где  $z_i \in \{0, 1\}, i = 0, 1, \dots, s-1$ , выполнено равенство  $Vec_s(z) = z_{s-1} \parallel \dots \parallel z_1 \parallel z_0$ ;
- $Int_s: V_s \rightarrow \mathbb{Z}_2^s$
- отображение, обратное к отображению  $Vec_s$ , т.е.  $Int_s = Vec_s^{-1}$ ;
- $k$
- параметр алгоритма блочного шифрования, называемый длиной ключа;
- $n$
- параметр алгоритма блочного шифрования, называемый длиной блока;

- $E: V_n \times V_k \rightarrow V_n$  – отображение, реализующее базовый алгоритм блочного шифрования и осуществляющее преобразование блока открытого текста  $P \in V_n$  с использованием ключа (шифрования)  $K \in V_k$  в блок шифртекста  $C \in V_n$ :  $E(P, K) = C$ ;
- $e_K: V_n \rightarrow V_n$  – отображение, реализующее зашифрование с использованием ключа  $K \in V_k$ , т.е.  $e_K(P) = E(P, K)$  для всех  $P \in V_n$ ;
- $d_K: V_n \rightarrow V_n$  – отображение, реализующее расшифрование с использованием ключа  $K \in V_k$ , т.е.  $d_K = e_K^{-1}$ .

### 3 Общие положения

Настоящий стандарт определяет следующие режимы работы алгоритмов блочного шифрования:

- режим простой замены (Electronic Codebook, ECB);
- режим гаммирования (Counter, CTR);
- режим гаммирования с обратной связью по выходу (Output Feedback, OFB);
- режим простой замены с сцеплением (Cipher Block Chaining, CBC);
- режим гаммирования с обратной связью по шифртексту (Cipher Feedback, CFB);
- режим выработки имитовставки (Message Authentication Code algorithm).

Данные режимы могут использоваться в качестве режимов для блочных шифров с произвольной длиной блока  $n$ .

## 4 Вспомогательные операции

### 4.1 Дополнение сообщения

Отдельные из описанных ниже режимов работы (режим гаммирования, режим гаммирования с обратной связью по выходу, режим гаммирования с обратной связью по шифртексту) могут осуществлять криптографическое преобразование сообщений произвольной длины. Для других режимов (режим простой замены,

режим простой замены с зацеплением) требуется, чтобы длина сообщения была кратна некоторой величине  $\ell$ . В последнем случае при работе с сообщениями произвольной длины необходимо применение процедуры дополнения сообщения до требуемой длины. Ниже приведены три процедуры дополнения.

Пусть  $P \in V^*$  исходное сообщение, подлежащее зашифрованию.

#### 4.1.1 Процедура 1

Пусть  $|P| \equiv r \pmod{\ell}$ . Положим

$$P^* = \begin{cases} P, & \text{если } r = 0, \\ P \parallel 0^{\ell-r}, & \text{иначе.} \end{cases}$$

**Примечание** – Описанная процедура в некоторых случаях не обеспечивает однозначного восстановления исходного сообщения. Например, результаты дополнения сообщений  $P_1$ , такого что  $|P_1| = \ell \cdot q - 1$  для некоторого  $q$ , и  $P_2 = P_1 \parallel 0$  будут одинаковы. В этом случае для однозначного восстановления необходимо дополнительно знать длину исходного сообщения.

#### 4.1.2 Процедура 2

Пусть  $|P| \equiv r \pmod{\ell}$ . Положим

$$P^* = P \parallel 1 \parallel 0^{\ell-r-1}.$$

**Примечание** – Данная процедура обеспечивает однозначное восстановление исходного сообщения. При этом если длина исходного сообщения кратна  $\ell$ , то длина дополненного сообщения будет увеличена.

#### 4.1.3 Процедура 3

Пусть  $|P| \equiv r \pmod{\ell}$ .

В зависимости от значения  $r$  возможны случаи:

- если  $r = n$ , то последний блок не изменяется  $P^* = P$ ;
- если  $r < n$ , то применяется процедура 2.

**Примечания**

1 Данная процедура обязательна для режима выработки имитовставки (5.6) и не рекомендуется для использования в других режимах (5.1 - 5.5).

2 Выбор конкретной процедуры дополнения предоставляется разработчику информационной системы и/или регламентируется другими нормативными документами.

## 4.2 Выработка начального значения

В некоторых режимах работы используются величины, начальное значение которых вычисляется на основании синхропосылки  $IV$ ; обозначим через  $m$  суммарную длину указанных величин. Будем обозначать процедуру выработки начального значения через  $I_m: V_{|M|} \rightarrow V_m$  и называть процедурой инициализации. Будем называть процедуру инициализации тривиальной, если  $I_{|M|} = IV$ . Если не оговорено иное, будем считать, что используется тривиальная процедура инициализации на основе синхропосылки необходимой длины.

Во всех описываемых в настоящем стандарте режимах работы не требуется обеспечение конфиденциальности синхропосылки. Вместе с тем процедура выработки синхропосылки должна удовлетворять одному из следующих требований.

- Значения синхропосылки для режимов простой замены с зацеплением и гаммирования с обратной связью по шифртексту необходимо выбирать случайно, равновероятно и независимо друг от друга из множества всех допустимых значений. В этом случае значение каждой используемой синхропосылки  $IV$  должно быть непредсказуемым (случайным или псевдослучайным): зная значения всех других используемых синхропосылок, значение  $IV$  нельзя определить с вероятностью большей, чем  $2^{-|M|}$ .
- Все значения синхропосылок, выработанных для зашифрования на одном и том же ключе в режиме гаммирования, должны быть уникальными, т.е. попарно различными. Для выработки значений синхропосылок может быть использован детерминированный счетчик.
- Значение синхропосылки для режима гаммирования с обратной связью по выходу должно быть либо непредсказуемым (случайным или псевдослучайным), либо уникальным.

**Примечание** – Режим простой замены не предусматривает использования синхропосылки.

## 4.3 Процедура усечения

В некоторых режимах используется усечение строк длины  $n$  до строк длины  $s$ ,  $s \leq n$ , с использованием функции  $T_s = \text{MSB}_s$ , т.е. в качестве операции усечения используется операция взятия бит с большими номерами.

## 5 Режимы работы алгоритмов блочного шифрования

### 5.1 Режим простой замены

Длина сообщений, зашифровываемых в режиме простой замены, должна быть кратна длине блока базового алгоритма блочного шифрования  $n$ , поэтому, при необходимости, к исходному сообщению должна быть предварительно применена процедура дополнения.

Зашифрование (расшифрование) в режиме простой замены заключается в зашифровании (расшифровании) каждого блока текста с помощью базового алгоритма блочного шифрования.

#### 5.1.1 Зашифрование

Открытый и, при необходимости, дополненный текст  $P \in V^*$ ,  $|P| = n \cdot q$ , представляется в виде:  $P = P_1 || P_2 || \dots || P_q$ ,  $P_i \in V_n$ ,  $i = 1, 2, \dots, q$ . Блоки шифртекста вычисляются по следующему правилу:

$$C_i = e_K(P_i), i = 1, 2, \dots, q. \quad (1)$$

Результирующий шифртекст имеет вид:

$$C = C_1 || C_2 || \dots || C_q.$$

Зашифрование в режиме простой замены проиллюстрировано на рисунке 1.

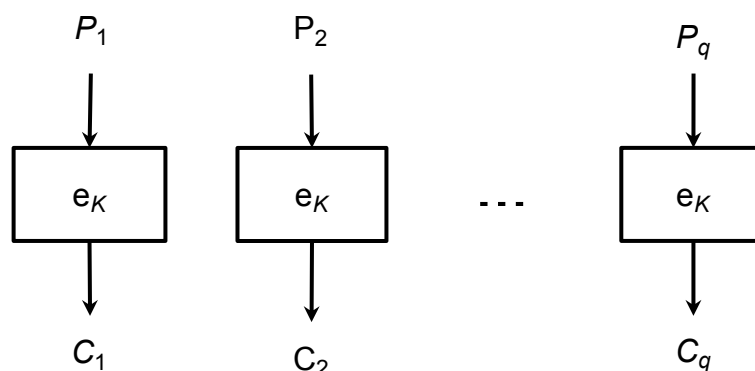


Рисунок 1 – Зашифрование в режиме простой замены

#### 5.1.2 Расшифрование

Шифртекст представляется в виде:  $C = C_1 || C_2 || \dots || C_q$ ,  $C_i \in V_n$ ,  $i = 1, 2, \dots, q$ . Блоки открытого текста вычисляются по следующему правилу:

$$P_i = d_K(C_i), i = 1, 2, \dots, q. \quad (2)$$

Исходный (дополненный) открытый текст имеет вид:

$$P = P_1 || P_2 || \dots || P_q.$$

**Примечание** – Если к исходному открытому тексту была применена процедура дополнения, то после расшифрования следует произвести обратную процедуру. Для однозначного восстановления сообщения может потребоваться знание длины исходного сообщения.

Расшифрование в режиме простой замены проиллюстрировано на рисунке 2.

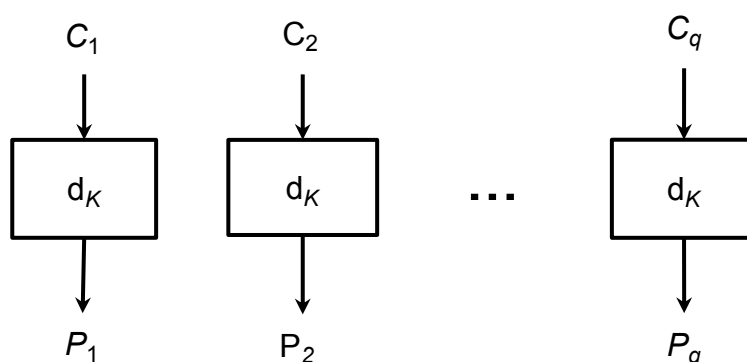


Рисунок 2 – Расшифрование в режиме простой замены

## 5.2 Режим гаммирования

Параметром режима гаммирования является целочисленная величина  $s$   $0 < s \leq n$ . При использовании режима гаммирования не требуется применение процедуры дополнения сообщения.

Для зашифрования (расшифрования) каждого отдельного открытого текста на одном ключе используется значение уникальной синхропосылки  $IV \in V_{\frac{n}{2}}$ .

Зашифрование в режиме гаммирования заключается в покомпонентном сложении открытого текста с гаммой шифра, которая вырабатывается блоками длины  $s$  путем зашифрования последовательности значений счетчика  $CTR_i \in V_n, i = 1, 2, \dots$ , базовым алгоритмом блочного шифрования с последующим усечением. Начальным значением счетчика является  $CTR_1 = I_n(IV) = IV || 0^{\frac{n}{2}}$ . Последующие значения счетчика вырабатываются с помощью функции  $Add: V_n \rightarrow V_n$  следующим образом:

$$CTR_{i+1} = Add(CTR_i) = Vec_n(Int_n(CTR_i) \boxplus_n 1). \quad (3)$$

### 5.2.1 Зашифрование

Открытый текст  $P \in V^*$  представляется в виде  $P = P_1 || P_2 || \dots || P_q$ ,  $P_i \in V_s$ ,  $i = 1, 2, \dots, q-1$ ,  $P_q \in V_r$ ,  $r \leq s$ . Блоки шифртекста вычисляются по следующему правилу:

$$\begin{cases} C_i = P_i \oplus T_s(e_K(CTR_i)), & i = 1, 2, \dots, q-1, \\ C_q = P_q \oplus T_r(e_K(CTR_q)). \end{cases} \quad (4)$$

Результирующий шифртекст имеет вид:

$$C = C_1 || C_2 || \dots || C_q.$$

Зашифрование в режиме гаммирования проиллюстрировано на рисунке 3.

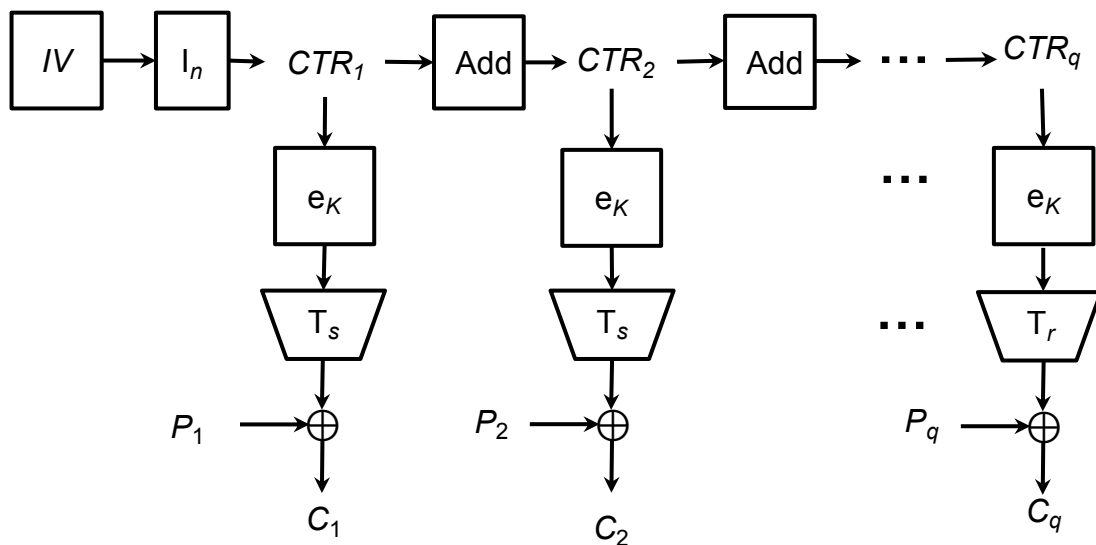


Рисунок 3 – Зашифрование в режиме гаммирования

### 5.2.2 Расшифрование

Шифртекст представляется в виде:  $C = C_1 || C_2 || \dots || C_q$ ,  $C_i \in V_s$ ,  $i = 1, 2, \dots, q-1$ ,  $C_q \in V_r$ ,  $r \leq s$ .

Блоки открытого текста вычисляются по следующему правилу:

$$\begin{cases} P_i = C_i \oplus T_s(e_K(CTR_i)), & i = 1, 2, \dots, q-1, \\ P_q = C_q \oplus T_r(e_K(CTR_q)). \end{cases} \quad (5)$$

Исходный открытый текст имеет вид

$$P = P_1 || P_2 || \dots || P_q.$$

Расшифрование в режиме гаммирования проиллюстрировано на рисунке 4.

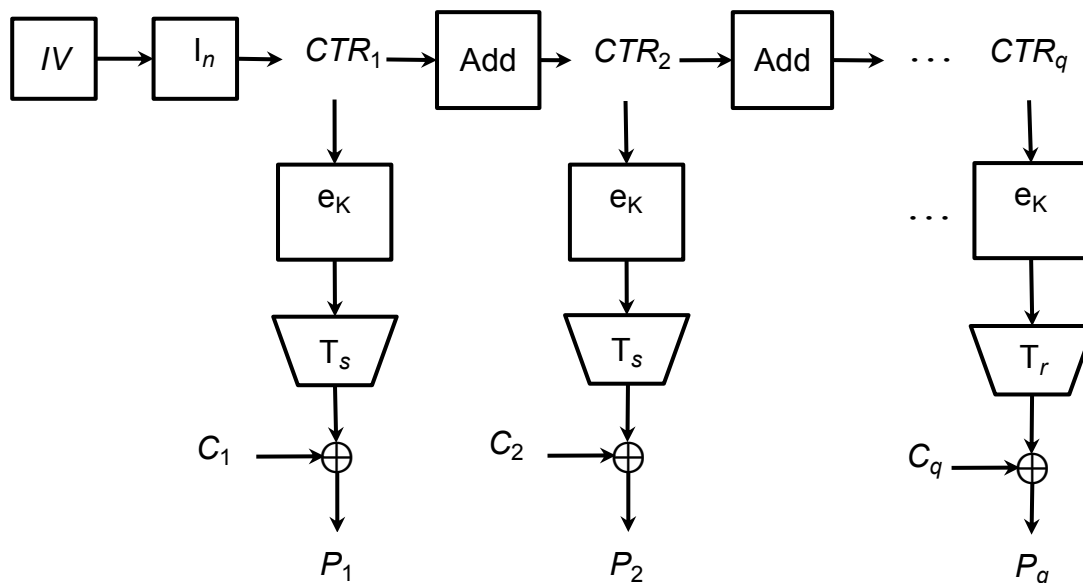


Рисунок 4 – Расшифрование в режиме гаммирования

### 5.3 Режим гаммирования с обратной связью по выходу

Параметрами режима гаммирования с обратной связью по выходу являются целочисленные величины  $s$  и  $m$ ,  $0 < s \leq n$ ,  $m = n \cdot z$ ,  $z \geq 1$  – целое число.

При использовании режима гаммирования с обратной связью по выходу не требуется применение процедуры дополнения сообщения.

При шифровании на одном ключе для каждого отдельного открытого текста используется значение уникальной или непредсказуемой (случайной или псевдослучайной) синхропосылки  $IV \in V_m$ .

При шифровании в режиме гаммирования с обратной связью по выходу используется двоичный регистр сдвига  $R$  длины  $m$ . Начальным заполнением регистра является значение синхропосылки  $IV$ .

Зашифрование в режиме гаммирования с обратной связью по выходу заключается в покомпонентном сложении открытого текста с гаммой шифра, которая вырабатывается блоками длины  $s$ . При вычислении очередного блока гаммы выполняется зашифрование  $n$  разрядов регистра сдвига с большими номерами базовым алгоритмом блочного шифрования. Затем заполнение регистра сдвигается на  $n$  бит в сторону разрядов с большими номерами, при этом в разряды с меньшими номерами записывается полученный выход базового алгоритма блочного



шифрования. Блок гаммы вычисляется путем усечения выхода базового алгоритма блочного шифрования.

### 5.3.1 Зашифрование

Открытый текст  $P \in V^*$  представляется в виде  $P = P_1 || P_2 || \dots || P_q$ ,  $P_i \in V_s$ ,  $i = 1, 2, \dots, q-1$ ,  $P_q \in V_r$ ,  $r \leq s$ . Блоки шифртекста вычисляются по следующему правилу:

$$\begin{aligned}
 R_1 &= IV, \\
 \begin{cases} Y_i = e_K(\text{MSB}_n(R_i)), \\ C_i = P_i \oplus T_s(Y_i), \\ R_{i+1} = \text{LSB}_{m-n}(R_i) || Y_i, \end{cases} & i = 1, 2, \dots, q-1, \quad (6) \\
 Y_q &= e_K(\text{MSB}_n(R_q)), \\
 C_q &= P_q \oplus T_r(Y_q).
 \end{aligned}$$

Результирующий шифртекст имеет вид:

$$C = C_1 || C_2 || \dots || C_q.$$

Зашифрование в режиме гаммирования с обратной связью по выходу проиллюстрировано на рисунке 5.

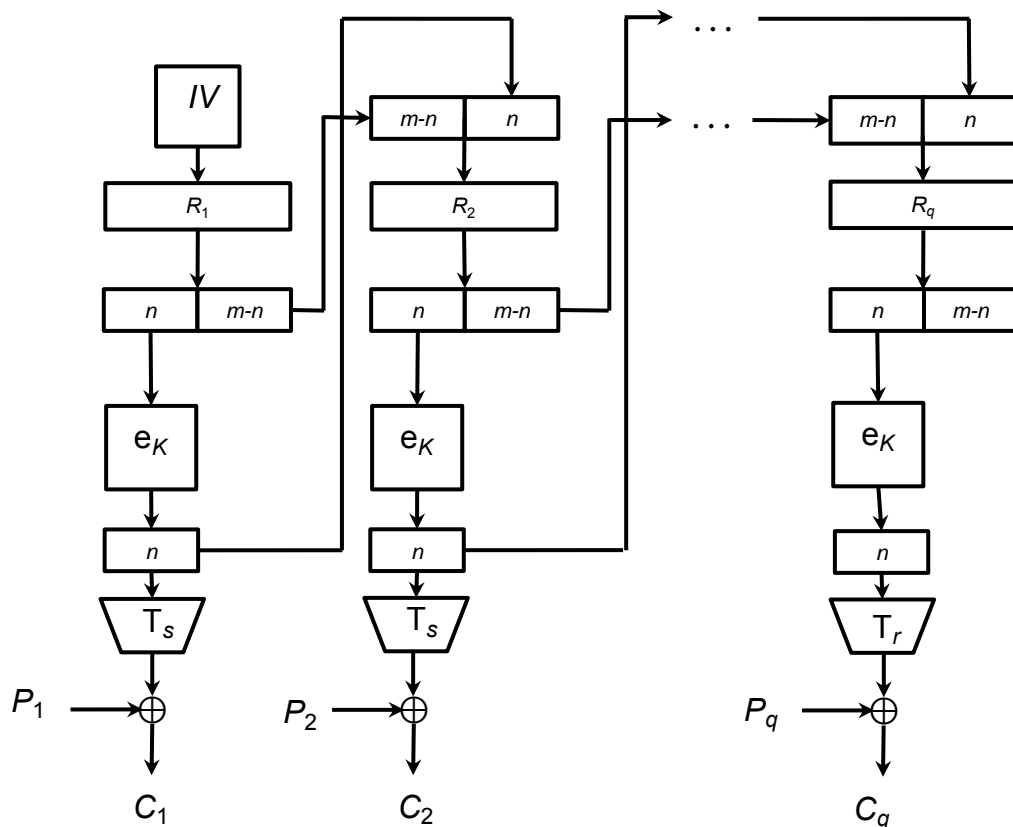


Рисунок 5 – Зашифрование в режиме гаммирования с обратной связью по выходу

### 5.3.2 Расшифрование

Шифртекст представляется в виде:  $C = C_1 || C_2 || \dots || C_q$ ,  $C_i \in V_s$ ,  $i = 1, 2, \dots, q-1$ ,  $C_q \in V_r$ ,  $r \leq s$ .

Блоки открытого текста вычисляются по следующему правилу:

$$\begin{aligned}
 R_1 &= IV, \\
 \begin{cases} Y_i = e_K(\text{MSB}_n(R_i)), \\ P_i = C_i \oplus T_s(Y_i), \\ R_{i+1} = \text{LSB}_{m-n}(R_i) || Y_i, \end{cases} & i = 1, 2, \dots, q-1, \\
 Y_q &= e_K(\text{MSB}_n(R_q)), \\
 P_q &= C_q \oplus T_r(Y_q)
 \end{aligned} \tag{7}$$

Исходный открытый текст имеет вид

$$P = P_1 || P_2 || \dots || P_q.$$

Расшифрование в режиме гаммирования с обратной связью по выходу проиллюстрировано на рисунке 6.

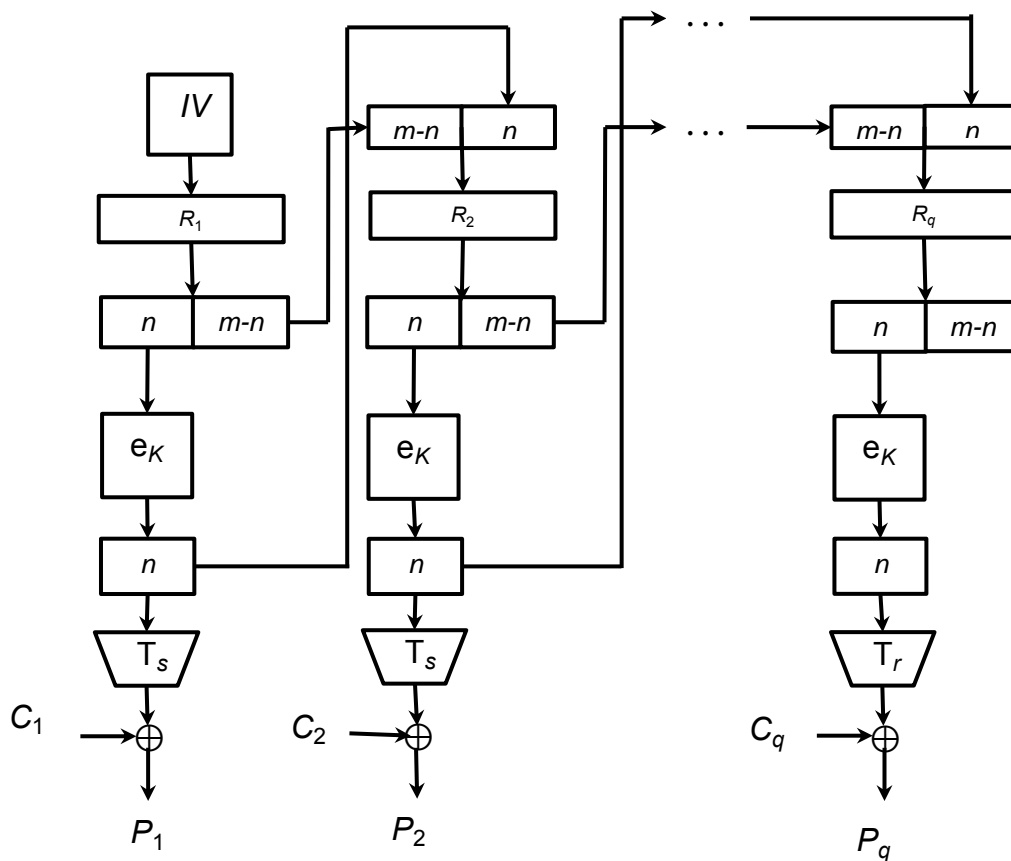


Рисунок 6 – Расшифрование в режиме гаммирования с обратной связью по выходу

#### 5.4 Режим простой замены с зацеплением

Параметром режима простой замены с зацеплением является целочисленная величина  $m$ ,  $m = n \cdot z$ ,  $z \geq 1$  – целое число.

Длина сообщений, зашифровываемых в режиме простой замены с зацеплением, должна быть кратна длине блока базового алгоритма блочного шифрования  $n$ , поэтому, при необходимости, к исходному сообщению должна быть предварительно применена процедура дополнения.

При шифровании на одном ключе для каждого отдельного открытого текста используется значение непредсказуемой (случайной или псевдослучайной) синхропосылки  $IV \in V_m$ .

При шифровании в режиме простой замены с зацеплением используется двоичный регистр сдвига  $R$  длины  $m$ . Начальным заполнением регистра является значение синхропосылки  $IV$ .

В режиме простой замены с зацеплением очередной блок шифртекста получается путем зашифрования результата покомпонентного сложения значения очередного блока открытого текста со значением  $n$  разрядов регистра сдвига с большими номерами. Затем регистр сдвигается на один блок в сторону разрядов с большими номерами. В разряды с меньшими номерами записывается значение блока шифртекста.

### 5.4.1 Зашифрование

Открытый и, при необходимости, дополненный текст  $P \in V^*$ ,  $|P| = n \cdot q$ , представляется в виде:  $P = P_1 || P_2 || \dots || P_q$ ,  $P_i \in V_n$ ,  $i = 1, 2, \dots, q$ . Блоки шифртекста вычисляются по следующему правилу:

$$\begin{aligned} R_1 &= IV, \\ \begin{cases} C_i = e_K(P_i \oplus \text{MSB}_n(R_i)), \\ R_{i+1} = \text{LSB}_{m-n}(R_i) || C_i, \end{cases} & i = 1, 2, \dots, q-1, \\ C_q &= e_K(P_q \oplus \text{MSB}_n(R_q)). \end{aligned} \quad (8)$$

Результирующий шифртекст имеет вид:

$$C = C_1 || C_2 || \dots || C_q.$$

Зашифрование в режиме простой замены с зацеплением проиллюстрировано на рисунке 7.

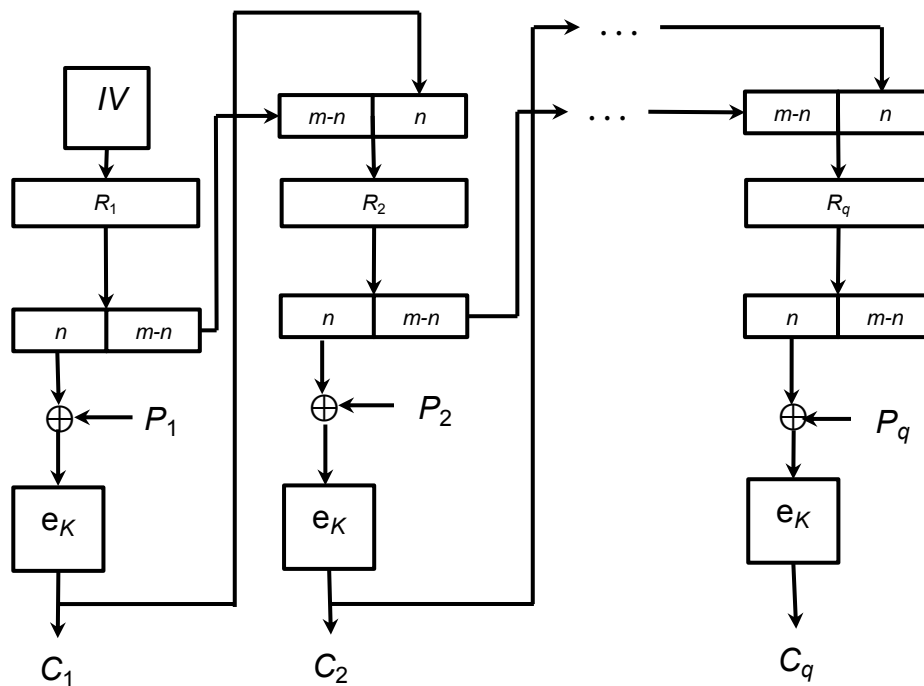


Рисунок 7 – Зашифрование в режиме простой замены с зацеплением

### 5.4.2 Расшифрование

Шифртекст представляется в виде:  $C = C_1 || C_2 || \dots || C_q$ ,  $C_i \in V_n$ ,  $i = 1, 2, \dots, q$ .

Блоки открытого текста вычисляются по следующему правилу:

$$\begin{aligned}
 R_1 &= IV, \\
 \begin{cases} P_i = d_K(C_i) \oplus \text{MSB}_n(R_i), \\ R_{i+1} = \text{LSB}_{m-n}(R_i) || C_i, \end{cases} & i = 1, 2, \dots, q-1, \\
 P_q &= d_K(C_q) \oplus \text{MSB}_n(R_q).
 \end{aligned} \tag{9}$$

Исходный (дополненный) открытый текст имеет вид:

$$P = P_1 || P_2 || \dots || P_q.$$

**Примечание** – Если к исходному открытому тексту была применена процедура дополнения, то после расшифрования следует произвести обратную процедуру. Для однозначного восстановления сообщения может потребоваться знание длины исходного сообщения.

Расшифрование в режиме простой замены с зацеплением проиллюстрировано на рисунке 8.

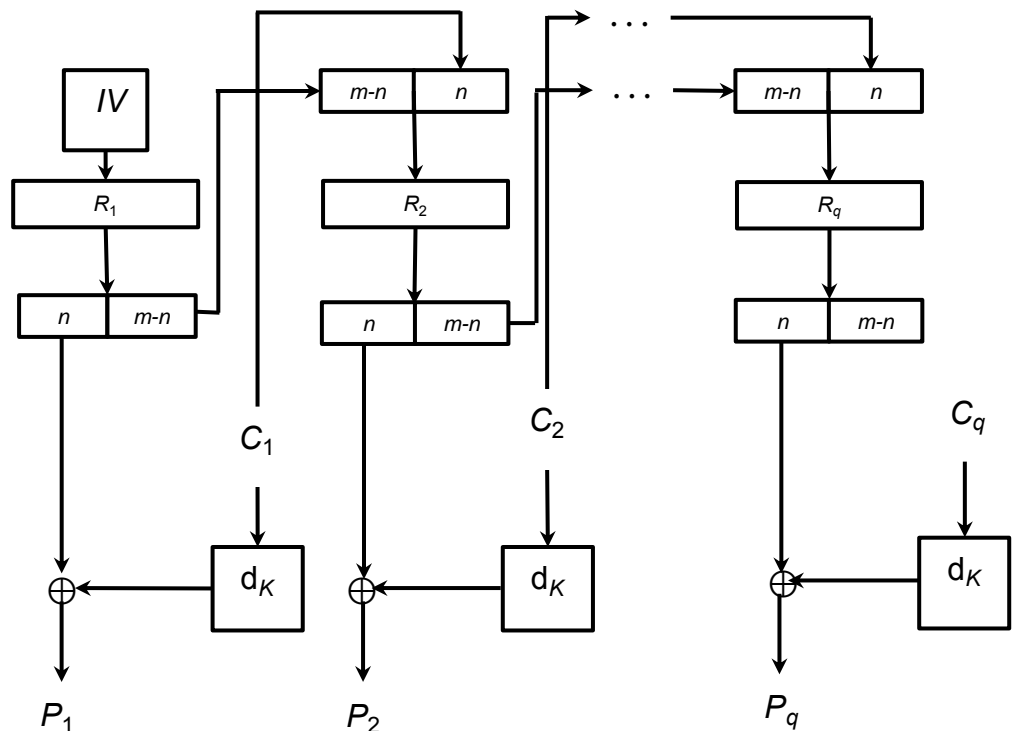


Рисунок 8 – Расшифрование в режиме простой замены с зацеплением

### 5.5 Режим гаммирования с обратной связью по шифртексту

Параметрами режима гаммирования с обратной связью по шифртексту являются целочисленные величины  $s$  и  $m$ ,  $0 < s \leq n$ ,  $n \leq m$ .

В конкретной системе обработки информации на длину сообщения  $P$  может как накладываться ограничение  $|P| = s \cdot q$ , так и не накладываться никаких ограничений. В случае если такое ограничение накладывается, к исходному сообщению, при необходимости, должна быть предварительно применена процедура дополнения.

При шифровании на одном ключе для каждого отдельного открытого текста используется значение непредсказуемой (случайной или псевдослучайной) синхросылки  $IV \in V_m$ .

При шифровании в режиме гаммирования с обратной связью по шифртексту используется двоичный регистр сдвига  $R$  длины  $m$ . Начальным заполнением регистра является значение синхросылки  $IV$ .

Зашифрование в режиме гаммирования с обратной связью по шифртексту заключается в покомпонентном сложении открытого текста с гаммой шифра, которая вырабатывается блоками длины  $s$ . При вычислении очередного блока гаммы выполняется зашифрование  $n$  разрядов регистра сдвига с большими номерами базовым алгоритмом блочного шифрования с последующим усечением. Затем заполнение регистра сдвигается на  $s$  разрядов в сторону разрядов с большими номерами, при этом в разряды с меньшими номерами записывается полученный блок шифртекста, являющийся результатом покомпонентного сложения гаммы шифра и блока открытого текста.

### 5.5.1 Зашифрование

Открытый текст  $P \in V^*$  представляется в виде  $P = P_1 || P_2 || \dots || P_q$ ,  $P_i \in V_s$ ,  $i = 1, 2, \dots, q-1$ ,  $P_q \in V_r$ ,  $r \leq s$ . Блоки шифртекста вычисляются по следующему правилу:

$$\begin{aligned} R_1 &= IV, \\ \begin{cases} C_i = P_i \oplus T_s(e_K(\text{MSB}_n(R_i))), \\ R_{i+1} = \text{LSB}_{m-s}(R_i) || C_i, \end{cases} & i = 1, 2, \dots, q-1, \quad (10) \\ C_q &= P_q \oplus T_r(e_K(\text{MSB}_n(R_q))). \end{aligned}$$

Результирующий шифртекст имеет вид:

$$C = C_1 || C_2 || \dots || C_q.$$

Зашифрование в режиме гаммирования с обратной связью по шифртексту проиллюстрировано на рисунке 9.

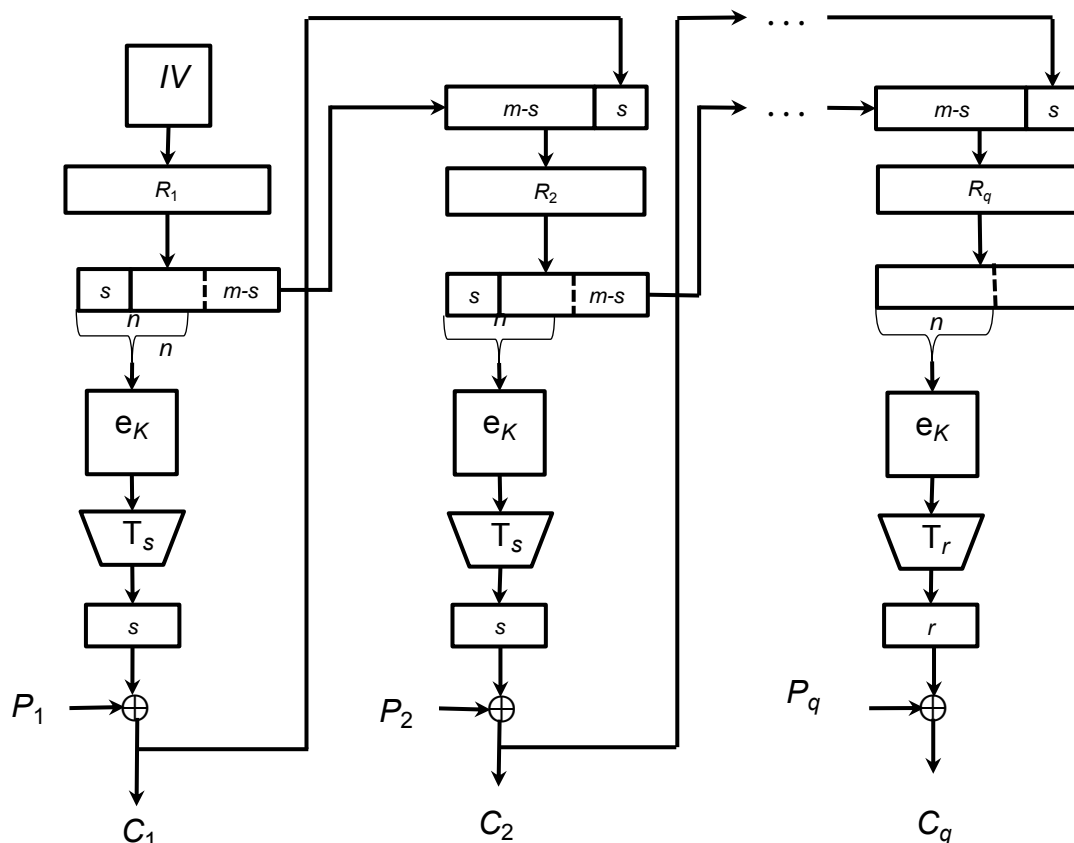


Рисунок 9 – Зашифрование в режиме гаммирования с обратной связью по шифртексту

### 5.5.2 Расшифрование

Шифртекст представляется в виде:  $C = C_1 || C_2 || \dots || C_q$ ,  $C_i \in V_s$ ,  $i = 1, 2, \dots, q-1$ ,  $C_q \in V_r$ ,  $r \leq s$ . Блоки открытого текста вычисляются по следующему правилу:

$$\begin{aligned}
 R_1 &= IV, \\
 \begin{cases} P_i = C_i \oplus T_s(e_K(\text{MSB}_n(R_i))), \\ R_{i+1} = \text{LSB}_{m-s}(R_i) || C_i, \end{cases} & i = 1, 2, \dots, q-1, \quad (11) \\
 P_q &= C_q \oplus T_r(e_K(\text{MSB}_n(R_q))).
 \end{aligned}$$

Исходный открытый текст имеет вид:

$$P = P_1 || P_2 || \dots || P_q.$$

**Примечание** – Если к исходному открытому тексту была применена процедура дополнения, то после расшифрования следует произвести обратную процедуру. Для



однозначного восстановления сообщения может потребоваться знание длины исходного сообщения.

Расшифрование в режиме гаммирования с обратной связью по шифртексту проиллюстрировано на рисунке 10.

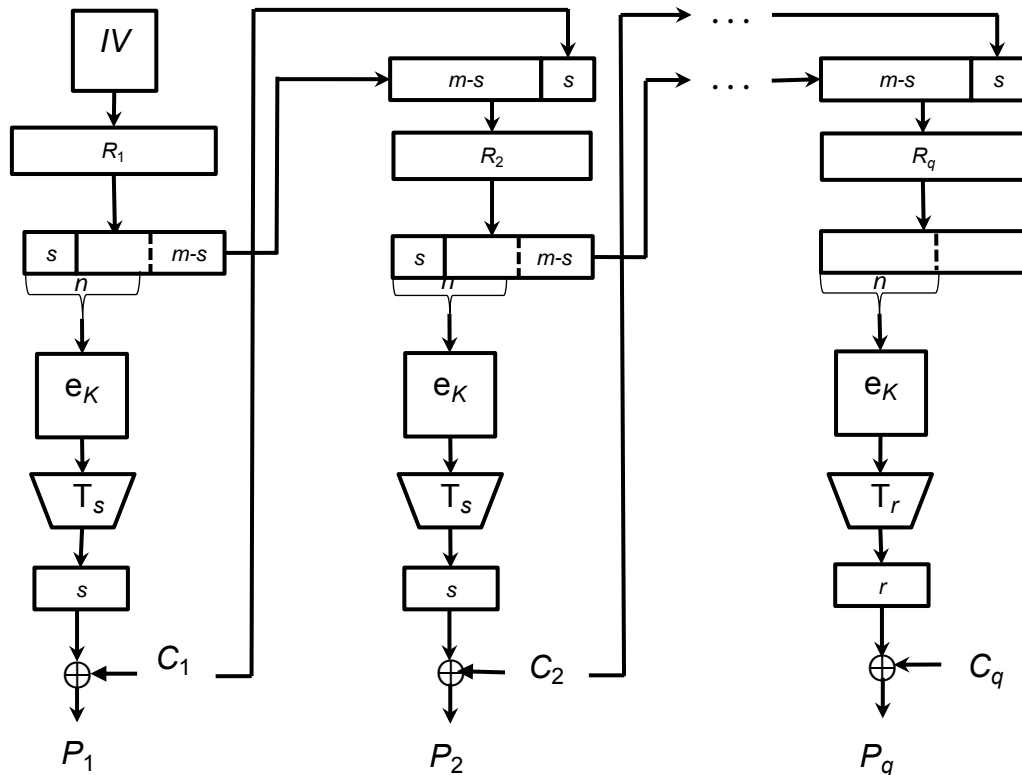


Рисунок 10 – Расшифрование в режиме гаммирования с обратной связью по шифртексту

## 5.6 Режим выработки имитовставки

Режим выработки имитовставки, описание которого представлено ниже, реализует конструкцию OMAC1 (стандартизован в ISO под названием CMAC [1]).

Параметром режима является длина имитовставки (в битах)  $0 < s \leq n$ .

### 5.6.1 Выработка вспомогательных ключей

При вычислении значения имитовставки используются вспомогательные ключи, которые вычисляются с использованием ключа  $K$ . Длины вспомогательных ключей равны длине блока  $n$  базового алгоритма блочного шифрования.

Процедура выработки вспомогательных ключей может быть представлена в следующей форме

$$\begin{aligned}
 R &= e_K(0^n); \\
 K_1 &= \begin{cases} R \ll 1, & \text{если } \text{MSB}_1(R) = 0, \\ ((R \ll 1) \oplus B_n), & \text{иначе;} \end{cases} \\
 K_2 &= \begin{cases} K_1 \ll 1, & \text{если } \text{MSB}_1(K_1) = 0, \\ ((K_1 \ll 1) \oplus B_n), & \text{иначе;} \end{cases}
 \end{aligned}$$

где  $B_{64} = 0^{59} \parallel 11011$ ,  $B_{128} = 0^{120} \parallel 10000111$ .

Если значение  $n$  отлично от 64 и 128, следует использовать следующую процедуру определения значения константы  $B_n$ . Рассмотрим множество примитивных многочленов степени  $n$  над полем  $\text{GF}(2)$  с наименьшим количеством ненулевых коэффициентов. Упорядочим это множество лексикографически по возрастанию векторов коэффициентов и обозначим через  $f_n(x)$  первый многочлен в этом упорядоченном множестве.

Рассмотрим поле  $\text{GF}(2^n)[x] / (f_n(x))$ , зафиксируем в нем степенной базис и будем обозначать операцию умножения в этом поле символом  $\otimes$ . Вспомогательные ключи  $K_1$  и  $K_2$  вычисляются следующим образом:

$$\begin{cases} R = e_K(0^n), \\ K_1 = \text{Poly}_n^{-1}(\text{Poly}_n(R) \otimes x), \\ K_2 = \text{Poly}_n^{-1}(\text{Poly}_n(R) \otimes x^2). \end{cases} \quad (12)$$

**Примечание** – Вспомогательные ключи  $K_1$ ,  $K_2$  и промежуточное значение  $R$  наряду с ключом  $K$  являются секретными параметрами. Компрометация какого-либо из этих значений приводит к возможности построения эффективных методов анализа всего алгоритма.

### 5.6.2 Вычисление значения имитовставки

Процедура вычисления значения имитовставки похожа на процедуру зашифрования в режиме простой замены с сцеплением при  $m = n$  и инициализации начального заполнения регистра сдвига значением  $0^n$ : на вход алгоритму шифрования подается результат покомпонентного сложения очередного блока текста и результата зашифрования на предыдущем шаге. Основное отличие заключается в процедуре обработки последнего блока: на вход базовому алгоритму

блочного шифрования подается результат покомпонентного сложения последнего блока, результата зашифрования на предыдущем шаге и одного из вспомогательных ключей. Конкретный вспомогательный ключ выбирается в зависимости от того, является ли последний блок исходного сообщения полным или нет. Значением имитовставки MAC является результат применения процедуры усечения к выходу алгоритма шифрования при обработке последнего блока.

Исходное сообщение  $P \in V^*$ , для которого требуется вычислить имитовставку, представляется в виде:

$$P = P_1 || P_2 || \dots || P_q,$$

где  $P_i \in V_n, i = 1, 2, \dots, q-1, P_q \in V_r, r \leq n$ .

Процедура вычисления имитовставки описывается следующим образом:

$$\begin{aligned} C_0 &= 0^n, \\ C_i &= e_K(P_i \oplus C_{i-1}), i = 1, 2, \dots, q-1, \\ MAC &= T_s(e_K(P_q^* \oplus C_{q-1} \oplus K^*)), \end{aligned} \quad (13)$$

где

$$K^* = \begin{cases} K_1, & \text{если } |P_q| = n, \\ K_2, & \text{иначе,} \end{cases}$$

$P_q^*$  – последний блок сообщения, полученного в результате дополнения исходного сообщения с помощью процедуры 3.

Процедура вычисления имитовставки проиллюстрирована на рисунках 11 – 13.

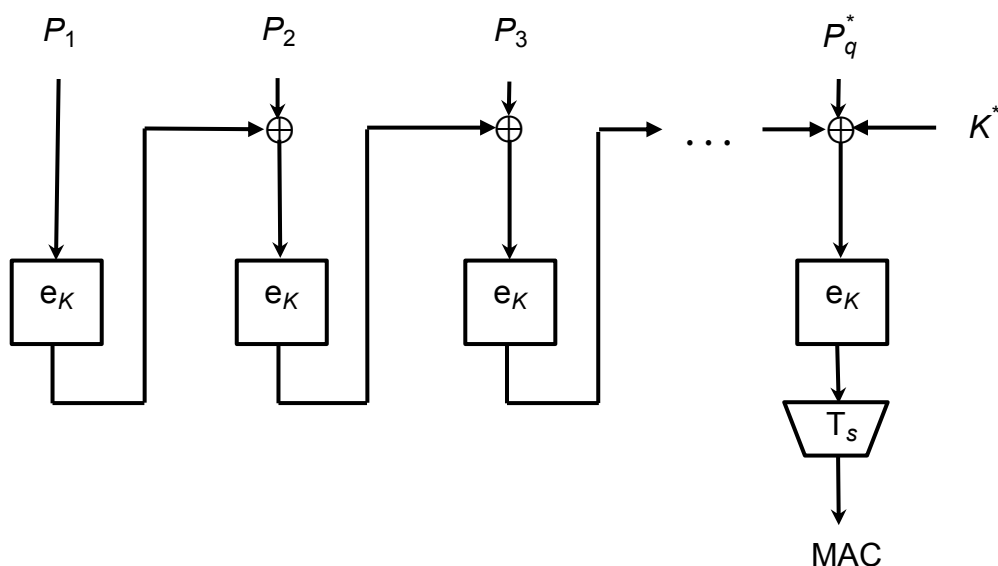


Рисунок 11 – Вычисление значения имитовставки – общий вид

П р и м е ч а н и е – Настоятельно рекомендуется не использовать ключ режима выработки имитовставки в других криптографических алгоритмах, в том числе в режимах, обеспечивающих конфиденциальность, описанных в 5.1 – 5.5.

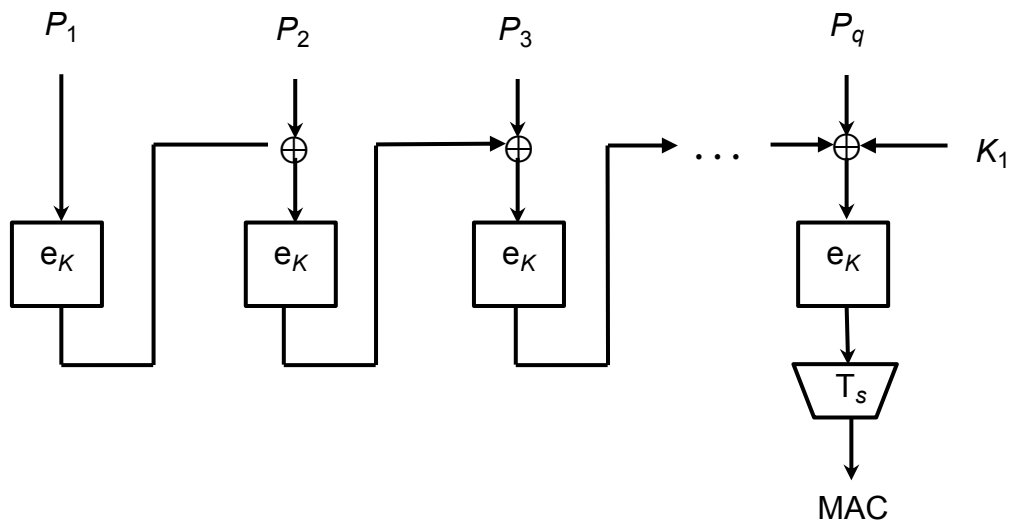


Рисунок 12 – Вычисление значения имитовставки – случай полного последнего блока

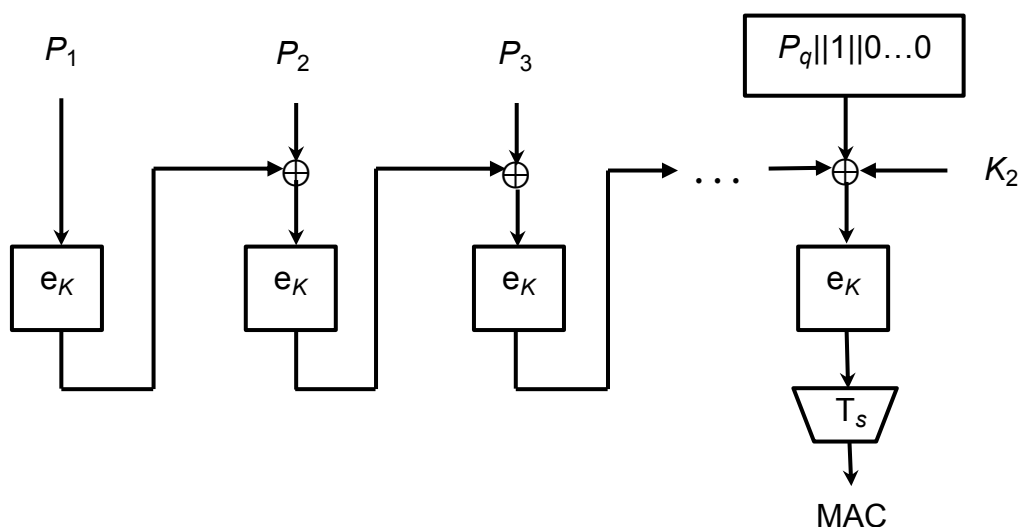


Рисунок 13 – Вычисление значения имитовставки – случай с дополнением последнего блока

## Приложение А

(справочное)

### Контрольные примеры

Данное приложение носит справочный характер и не является частью настоящего стандарта.

В данном приложении содержатся примеры для зашифрования и расшифрования сообщений, а также выработки имитовставки, с использованием режимов работы шифра, определенных в данном стандарте. Параметр  $s$  выбран равным  $n$  с целью упрощения проводимых вычислений, а параметр  $m$  выбирался из соображений демонстрации особенностей каждого режима шифрования. Двоичные строки из  $V^*$ , длина которых кратна 4, записываются в шестнадцатеричном виде, а символ конкатенации ("||") опускается. То есть, строка  $a \in V_{4r}$  будет представлена в виде  $a_{r-1}a_{r-2}\dots a_0$ , где  $a_i \in \{0, 1, \dots, 9, a, b, c, d, e, f\}$ ,  $i = 0, 1, \dots, r - 1$ .

В А.1 приведены примеры для блочного шифра с длиной блока  $n = 128$  бит («Кузнечик»). В А.2 приведены примеры для блочного шифра с длиной блока  $n = 64$  бит («Магма»).

#### А.1 Блочный шифр с длиной блока $n = 128$ бит

Примеры используют следующие параметры:

Ключ

$K = 8899aabbccddeeff0011223344556677fedcba98765432100123456789abcdef.$

Открытый текст – четыре 128-битных блока:

$P_1 = 1122334455667700feeddccbbaa9988,$

$P_2 = 00112233445566778899aabbccceeff0a,$

$P_3 = 112233445566778899aabbccceeff0a00,$

$P_4 = 2233445566778899aabbccceeff0a0011.$

### А.1.1 Режим простой замены

Т а б л и ц а А.1 – Зашифрование в режиме простой замены

Открытый текст	Шифртекст
1122334455667700feeddccbbaa9988	7f679d90bebc24305a468d42b9d4edcd
00112233445566778899aabbccceeff0a	b429912c6e0032f9285452d76718d08b
112233445566778899aabbccceeff0a00	f0ca33549d247ceef3f5a5313bd4b157
2233445566778899aabbccceeff0a0011	d0b09ccde830b9eb3a02c4c5aa8ada98

### А.1.2 Режим гаммирования

#### А.1.2.1 Зашифрование

$$s = n = 128,$$

$$IV = 1234567890abcef0.$$

Т а б л и ц а А.2 – Зашифрование в режиме гаммирования

$i$	1	2
$P_i$	1122334455667700feeddccbbaa9988	00112233445566778899aabbccceeff0a
Входной блок	1234567890abcef0000000000000000	1234567890abcef00000000000000001
Выходной блок	e0b7ebfa9468a6db2a95826efb173830	85ffc500b2f4582a7ba54e08f0ab21ee
$C_i$	f195d8bec10ed1dbd57b5fa240bda1b8	85eee733f6a13e5df33ce4b33c45dee4

*окончание таблицы А.2*

$i$	3	4
$P_i$	112233445566778899aabbccceeff0a00	2233445566778899aabbccceeff0a0011
Входной блок	1234567890abcef00000000000000002	1234567890abcef00000000000000003
Выходной блок	b4c8dbcfb353195b4c42cc3ddb9ba9a5	e9a2bee4947b322f7b7d1db6dfb7ba62
$C_i$	a5eae88be6356ed3d5e877f13564a3a5	cb91fab1f20cbab6d1c6d15820bdba73

#### А.1.2.2 Расшифрование

С использованием приведенных значений  $K$ ,  $IV$  и  $C$  с помощью операции расшифрования воспроизводятся исходные значения  $P_1, P_2, P_3, P_4$ .

### А.1.3 Режим гаммирования с обратной связью по выходу

#### А.1.3.1 Зашифрование

$$s = n = 128, m = 2n = 256,$$

$$IV = 1234567890abcef0a1b2c3d4e5f0011223344556677889901213141516171819.$$

Т а б л и ц а А.3 – Зашифрование в режиме гаммирования с обратной связью по выходу

$i$	1	2
$P_i$	1122334455667700feeddccbbaa9988	00112233445566778899aabbccceeff0a
Входной блок	1234567890abcef0a1b2c3d4e5f00112	23344556677889901213141516171819
Выходной блок	90a2391de4e25c2400f1a49232d0241d	ed4a659440d99cc3072c8b8d517dd9b5
$C_i$	81800a59b1842b24ff1f795e897abd95	ed5b47a7048cfab48fb521369d9326bf

*окончание таблицы А.3*

$i$	3	4
$P_i$	112233445566778899aabbccceeff0a00	2233445566778899aabbccceeff0a0011
Входной блок	90a2391de4e25c2400f1a49232d0241d	ed4a659440d99cc3072c8b8d517dd9b5
Выходной блок	778064e869c6cf3951a55c30fed78013	020dff9500640ef90a92eead099a3141
$C_i$	66a257ac3ca0b8b1c80fe7fc10288a13	203ebbc066138660a0292243f6903150

#### А.1.3.2 Расшифрование

С использованием приведенных значений  $K$ ,  $IV$  и  $C$  с помощью операции расшифрования воспроизводятся исходные значения  $P_1, P_2, P_3, P_4$ .

## А.1.4 Режим простой замены с зацеплением

### А.1.4.1 Зашифрование

$$m = 2n = 256,$$

$$IV = 1234567890abcef0a1b2c3d4e5f0011223344556677889901213141516171819.$$

Т а б л и ц а А.4 – Зашифрование в режиме простой замены с зацеплением

$i$	1	2
$P_i$	1122334455667700feeddccbbaa9988	00112233445566778899aabbccceeff0a
Входной блок	0316653cc5cdb9f05e5c1e185e5a989a	23256765232defe79a8abeaedaf9e713
Выходной блок	689972d4a085fa4d90e52e3d6d7dcc27	2826e661b478eca6af1e8e448d5ea5ac
$C_i$	689972d4a085fa4d90e52e3d6d7dcc27	2826e661b478eca6af1e8e448d5ea5ac

*окончание таблицы А.4*

$i$	3	4
$P_i$	112233445566778899aabbccceeff0a00	2233445566778899aabbccceeff0a0011
Входной блок	79bb4190f5e38dc5094f95f18382c627	0a15a234d20f643f05a542aa7254a5bd
Выходной блок	fe7babf1e91999e85640e8b0f49d90d0	167688065a895c631a2d9a1560b63970
$C_i$	fe7babf1e91999e85640e8b0f49d90d0	167688065a895c631a2d9a1560b63970

### А.1.4.2 Расшифрование

С использованием приведенных значений  $K$ ,  $IV$  и  $C$  с помощью операции расшифрования воспроизводятся исходные значения  $P_1, P_2, P_3, P_4$ .



## А.1.5 Режим гаммирования с обратной связью по шифртексту

### А.1.5.1 Зашифрование

$$s = n = 128, m = 2n = 256,$$

$IV = 1234567890abcef0a1b2c3d4e5f0011223344556677889901213141516171819.$

Т а б л и ц а А.5 – Зашифрование в режиме гаммирования с обратной связью по шифртексту

$i$	1	2
$P_i$	1122334455667700feeddccbbaa9988	00112233445566778899aabbccceeff0a
Входной блок	1234567890abcef0a1b2c3d4e5f00112	23344556677889901213141516171819
Выходной блок	90a2391de4e25c2400f1a49232d0241d	ed4a659440d99cc3072c8b8d517dd9b5
$C_i$	81800a59b1842b24ff1f795e897abd95	ed5b47a7048cfab48fb521369d9326bf

*окончание таблицы А.5*

$i$	3	4
$P_i$	112233445566778899aabbccceeff0a00	2233445566778899aabbccceeff0a0011
Входной блок	81800a59b1842b24ff1f795e897abd95	ed5b47a7048cfab48fb521369d9326bf
Выходной блок	68d09baf09a0fab01d879d82795d32b5	6dcdfa9828e5a57f6de01533bbf1f4c0
$C_i$	79f2a8eb5cc68d38842d264e97a238b5	4ffebeecd4e922de6c75bd9dd44fbf4d1

### А.1.5.2 Расшифрование

С использованием приведенных значений  $K$ ,  $IV$  и  $C$  с помощью операции расшифрования воспроизводятся исходные значения  $P_1, P_2, P_3, P_4$ .

## А.1.6 Режим выработки имитовставки

### А.1.6.1 Выработка вспомогательных ключей

$$R = 94bec15e269cf1e506f02b994c0a8ea0,$$

$$MSB_1(R) = 1,$$

$$K_1 = R \ll 1 \oplus B_n = 297d82bc4d39e3ca0de0573298151d40 \oplus 87 =$$

$$= 297d82bc4d39e3ca0de0573298151dc7,$$

$$MSB_1(K_1) = 0,$$

$$K_2 = K_1 \ll 1 = 297d82bc4d39e3ca0de0573298151dc7 \ll 1 =$$

$$= 52fb05789a73c7941bc0ae65302a3b8e,$$

$$|P_4| = n, K^* = K_1.$$

### А.1.6.2 Вычисление имитовставки

$$s = 64.$$

Т а б л и ц а А.6 – Вычисление имитовставки

$i$	1	2
$P_i$	1122334455667700feeddccbbaa9988	00112233445566778899aabbccceeff0a
Входной блок	1122334455667700feeddccbbaa9988	7f76bfa3fae94247d2df27f9753a12c7
Выходной блок	7f679d90bebc24305a468d42b9d4edcd	1ac9d976f83636f55ae9ef305e7c90d2

*окончание таблицы А.6*

$i$	3	4
$P_i$	112233445566778899aabbccceeff0a00	2233445566778899aabbccceeff0a0011
Входной блок	0bebea32ad50417dc34354fcb0839ad2	1e2a9c1d8cc03bfa0cb340971252fe24
Выходной блок	15645af4a78e50a9abe8db4b754de3f2	336f4d296059fbe34ddeb35b37749c67

$$MAC = 336f4d296059fbe3.$$

## А.2 Блочный шифр с длиной блока $n = 64$ бит

Примеры используют следующие параметры.

Ключ

$K = \text{ffeeddccbbaa99887766554433221100f0f1f2f3f4f5f6f7f8f9fafbfcfdfeff}$ .

Открытый текст – четыре 64-битных блока:

$P_1 = \text{92def06b3c130a59}$ ,

$P_2 = \text{db54c704f8189d20}$ ,

$P_3 = \text{4a98fb2e67a8024c}$ ,

$P_4 = \text{8912409b17b57e41}$ .

### А.2.1 Режим простой замены

Т а б л и ц а А.7 – Зашифрование в режиме простой замены

Открытый текст	Шифртекст
92def06b3c130a59	2b073f0494f372a0
db54c704f8189d20	de70e715d3556e48
4a98fb2e67a8024c	11d8d9e9eacfb1e
8912409b17b57e41	7c68260996c67efb

## А.2.2 Режим гаммирования

### А.2.2.1 Зашифрование

$$s = n = 64,$$

$$IV = 12345678.$$

Т а б л и ц а А.8 – Зашифрование в режиме гаммирования

$i$	1	2
$P_i$	92def06b3c130a59	db54c704f8189d20
Входной блок	1234567800000000	1234567800000001
Выходной блок	dc46e167aba4b365	e571ca972ef0c049
$C_i$	4e98110c 97b7b93c	3e250d93d6e85d69

окончание таблицы А.8

$i$	3	4
$P_i$	4a98fb2e67a8024c	8912409b17b57e41
Входной блок	1234567800000002	1234567800000003
Выходной блок	59f57da6601ad9a3	df9cf61bbce7df6c
$C_i$	136d868807b2dbef	568eb680ab52a12d

### А.2.2.2 Расшифрование

С использованием  $K$ ,  $IV$  и  $C$  с помощью операции расшифрования производятся исходные значения  $P_1, P_2, P_3, P_4$ .

## А.2.3 Режим гаммирования с обратной связью по выходу

### А.2.3.1 Зашифрование

$$s = n = 64, m = 2n = 128,$$

$$IV = 1234567890abcdef234567890abcdef1.$$

Т а б л и ц а А.9 – Зашифрование в режиме гаммирования с обратной связью

$i$	1	2
$P_i$	92def06b3c130a59	db54c704f8189d20
Входной блок	1234567890abcdef	234567890abcdef1
Выходной блок	49e910895a8336da	d612a348e78295bc
$C_i$	db37e0e266903c83	0d46644c1f9a089c

*окончание таблицы А.9*

$i$	3	4
$P_i$	4a98fb2e67a8024c	8912409b17b57e41
Входной блок	49e910895a8336da	d612a348e78295bc
Выходной блок	ea60cb4c24a63032	4136af23aafaa544
$C_i$	a0f83062430e327e	c824efb8bd4fdb05

### А.2.3.2 Расшифрование

С использованием приведенных значений  $K$ ,  $IV$  и  $C$  с помощью операции расшифрования воспроизводятся исходные значения  $P_1, P_2, P_3, P_4$ .

## А.2.4 Режим простой замены с зацеплением

### А.2.4.1 Зашифрование

$$m = 3n = 192,$$

$IV = 1234567890abcdef234567890abcdef134567890abcdef12.$

Т а б л и ц а А.10 – Зашифрование в режиме простой замены с зацеплением

$i$	1	2
$P_i$	92def06b3c130a59	db54c704f8189d20
Входной блок	80eaa613acb8c7b6	f811a08df2a443d1
Выходной блок	96d1b05eea683919	aff76129abb937b9
$C_i$	96d1b05eea683919	aff76129abb937b9
<i>окончание таблицы А.10</i>		
$i$	3	4
$P_i$	4a98fb2e67a8024c	8912409b17b57e41
Входной блок	7ece83becc65ed5e	1fc3f0c5fddd4758
Выходной блок	5058b4a1c4bc0019	20b78b1a7cd7e667
$C_i$	5058b4a1c4bc0019	20b78b1a7cd7e667

### А.2.4.2 Расшифрование

С использованием приведенных значений  $K$ ,  $IV$  и  $C$  с помощью операции расшифрования воспроизводятся исходные значения  $P_1, P_2, P_3, P_4$ .

## А.2.5 Режим гаммирования с обратной связью по шифртексту

### А.2.5.1 Зашифрование

$$s = n = 64, m = 2n = 128,$$

$$IV = 1234567890abcdef234567890abcdef1.$$

Т а б л и ц а А.11 – Зашифрование в режиме гаммирования с обратной связью по шифртексту

$i$	1	2
$P_i$	92def06b3c130a59	db54c704f8189d20
Входной блок	1234567890abcdef	234567890abcdef1
Выходной блок	49e910895a8336da	d612a348e78295bc
$C_i$	db37e0e266903c83	0d46644c1f9a089c

*окончание таблицы А.11*

$i$	3	4
$P_i$	4a98fb2e67a8024c	8912409b17b57e41
Входной блок	db37e0e266903c83	0d46644c1f9a089c
Выходной блок	6e25292d34bdd1c7	35d2728f36b22b44
$C_i$	24bdd2035315d38b	bcc0321421075505

### А.2.5.1. Расшифрование

С использованием приведенных значений  $K$ ,  $IV$  и  $C$  с помощью операции расшифрования воспроизводятся исходные значения  $P_1, P_2, P_3, P_4$ .

## А.2.6 Режим выработки имитовставки

### А.2.6.1 Выработка вспомогательных ключей

$R = 2fa2cd99a1290a12,$

$MSB_1(R) = 0, K_1 = R \ll 1 = 5f459b3342521424,$

$MSB_1(K_1) = 0,$  следовательно  $K_2 = K_1 \ll 1 = be8b366684a42848,$

$|P_4| = n, K^* = K_1.$

### А.2.6.2 Вычисление имитовставки

$s = 32.$

Т а б л и ц а А.12 – Вычисление имитовставки

$i$	1	2
$P_i$	92def06b3c130a59	db54c704f8189d20
Входной блок	92def06b3c130a59	f053f8006cebef80
Выходной блок	2b073f0494f372a0	c89ed814fd5e18e9

*окончание таблицы А.12*

$i$	3	4
$P_i$	4a98fb2e67a8024c	8912409b17b57e41
Входной блок	8206233a9af61aa5	216e6a2561cff165
Выходной блок	f739b18d34289b00	154e72102030c5bb

MAC = 154e7210.



## Библиография\*

- [1] ИСО/МЭК 9797-1:2011  
(ISO 9797-1:2011) Информационные технологии. Методы защиты. Коды аутентификации сообщений (MAC). Часть 1. Механизмы, использующие блочный шифр (Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher)
- [2] ИСО/МЭК 10116:2006  
(ISO/IEC 10116:2006) Информационные технологии. Методы обеспечения безопасности. Режимы работы для *n*-битовых блочных шифров (Information technology – Security techniques – Modes of operation for an *n*-bit block cipher)
- [3] ИСО/МЭК 10118-1:2000  
(ISO/IEC 10118-1:2000) Информационные технологии. Методы защиты информации. Хэш-функции. Часть 1. Общие положения (Information technology – Security techniques – Hash-functions – Part 1: General)
- [4] ИСО/МЭК 18033-1:2005  
(ISO/IEC 18033-1:2005) Информационные технологии. Методы и средства обеспечения безопасности. Алгоритмы шифрования. Часть 1. Общие положения (Information technology – Security techniques – Encryption algorithms – Part 1: General)
- [5] ИСО/МЭК 14888-1:2008  
(ISO/IEC 14888-1:2008) Информационные технологии. Методы защиты. Цифровые подписи с приложением. Часть 1. Общие положения (Information technology – Security techniques – Digital signatures with appendix – Part 1: General)

---

\* Оригиналы международных стандартов ИСО/МЭК находятся во ФГУП «Стандартинформ» Федерального агентства по техническому регулированию и метрологии.

---

УДК 681.3.06:006.354

ОКС 35.040

ОКСТУ 5002

П85

Ключевые слова: информационная технология, криптографическая защита информации, блочный шифр, режимы работы блочного шифра, конфиденциальность, целостность, имитовставка, гаммирование, зацепление

---