Министерство образования Республики Беларусь
Белорусский государственный университет
НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ ИНСТИТУТ
ПРИКЛАДНЫХ ПРОБЛЕМ МАТЕМАТИКИ И ИНФОРМАТИКИ

УТВЕРЖДАЮ
Директор НИИ прикладных проблем
математики и информатики

Ю.С.Харин
«___» _____ 2022 г.

МЕТОДИКА ИСПЫТАНИЙ СРЕДСТВ КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ ИНФОРМАЦИИ НА СООТВЕТСТВИЕ ТРЕБОВАНИЯМ СТБ 34.101.31-2020

МИ.10131.10.01

Листов 48

Минск 2022

Предисловие

Настоящая методика испытаний предназначена для использования в испытательных лабораториях при проведении сертификационных испытаний средств криптографической защиты информации на соответствие требованиям СТБ 34.101.31-2020 «Информационные технологии и безопасность. Алгоритмы шифрования и контроля целостности».

Содержание

1 Нормативные ссылки

В настоящем документе использованы ссылки на следующие стандарты:

ГОСТ 19.202-78 «Единая система программной документации. Спецификация. Требования к содержанию и оформлению».

ГОСТ 19.401-2000 «Единая система программной документации. Текст программы. Требования к содержанию, оформлению и контролю качества».

ГОСТ 19.402-2000 «Единая система программной документации. Описание программы. Требования к содержанию, оформлению и контролю качества».

ГОСТ 19.504-79 «Единая система программной документации. Руководство программиста. Требования к содержанию и оформлению».

СТБ 34.101.31-2020 «Информационные технологии и безопасность. Алгоритмы шифрования и контроля целостности».

СТБ 34.101.77-2020 «Информационные технологии и безопасность. Криптографические алгоритмы на основе sponge-функции».

2 Термины, обозначения и сокращения

В настоящем документе применяются термины и обозначения СТБ 34.101.31, а также следующие сокращения:

ЕСПД единая система программной документации;

СКЗИ средство криптографической защиты информации.

3 Объект и цель испытаний

На испытания представляется средство криптографической защиты информации (СКЗИ), реализующее криптографические алгоритмы СТБ 34.101.31, и документация на СКЗИ.

Целью испытаний является проверка соответствия объекта испытаний требованиям СТБ 34.101.31.

4 Требования к объекту испытаний

К программе объекта испытаний предъявляются следующие требования, подлежащие проверке во время проведения испытаний:

– в программе должны быть точно и полно реализовываны криптографические алгоритмы СТБ 34.101.31, поддерживаемые объектом испытаний;

– программа, реализующая криптографические алгоритмы и требования СТБ 34.101.31, не должна содержать недокументированные возможности.

Документация на объект испытаний должна включать документы «Спецификация», «Текст программы» и может включать документы «Описание программы», «Руководство

программиста» и другие документы. Документация может быть разработана в соответствии с требованиями единой системы программной документации (ЕСПД).

5 Средства и порядок испытаний

5.1 Общие сведения

Испытания программы состоят из трех этапов:

1 Анализ документации.

2 Тестирование программы.

3 Анализ исходных текстов программы.

Выполнение этапа 1 осуществляется экспертами по анализу документации, выполнение этапа 2 — экспертами по тестированию, а выполнение этапа 3 — экспертами по анализу исходных текстов. К проведению испытаний должно быть привлечено не менее двух экспертов по анализу исходных текстов и один или более эксперт по тестированию. К анализу документации должен быть привлечен, по крайней мере, один эксперт по анализу исходных текстов программ.

По результатам выполнения этапа испытаний эксперт оформляет протокол результатов проверок: протокол анализа документации, протокол тестирования, протокол анализа исходных текстов. В протоколе эксперт делает вывод о соответствии (не соответствии) программы требованиям СТБ 34.101.31. Если программа не поддерживает некоторые алгоритмы, определенные в СТБ 34.101.31, то в протоколе делается соответствующее примечание. Примеры оформления протоколов приводятся в приложениях А, Б, В. Допускается оформления протоколов в иной форме, но с обязательным указанием результатов по каждой проводимой проверке и вывода о соответствии (не соответствии).

Если в испытуемой программе используются реализации алгоритмов СТБ 34.101.31, которые в составе других программ имеют действующие сертификаты соответствия требованиям СТБ 34.101.31, то проверки по тестированию и анализу исходных текстов для данных реализаций могут не проводиться. При этом для подтверждения соответствия объекта испытаний требованиям СТБ 34.101.31 экспертом оформляется протокол проверки совпадения контрольных характеристик (хэш-значений) файлов реализации испытуемой программы с контрольными характеристиками соответствующих файлов, указанными в сертификатах соответствия.

На основании протоколов результатов проверок оформляется протокол испытаний, обобщающий результаты испытаний программы. В протоколе испытаний вывод о соответствии программы требованиям СТБ 34.101.31 делается тогда и только тогда, когда вывод о соответствии содержится во всех протоколах результатов проверок. Оформление протокола испытаний проводится в соответствии с требованиями технических нормативно-правовых актов в области сертификации продукции, а также документации, применяемой в испытательной лаборатории.

5.2 Анализ документации

Эксперт проводит анализ документации путем проверки соответствия документации программе объекта испытаний. Такой анализ состоит в получении экспертных заключений, касающихся проверки следующих документов:

– спецификация (см. п. 6.1.1);

– текст программы (см. п. 6.1.2);

– описание программы (см. п. 6.1.3);
– руководство программиста (см. п. 6.1.4).

Анализ документов «Описание программы» и «Руководство программиста» производится в случае их наличия.

5.3 Тестирование

Эксперт проводит тестирование путем выполнения испытуемой программы для некоторого набора проверочных входных значений и сравнения полученных результатов с истинными. Истинные результаты, используемые при тестировании, формируются с помощью эталонной реализации.

Эталонной считается реализация, которая ранее успешно прошла сертификационные испытания на соответствие СТБ 34.101.31 или которая удовлетворяет следующим условиям:

1 Проведен анализ исходных текстов программ эталонной реализации. К анализу привлекались, по меньшей мере, два независимых эксперта. Использовалась методика анализа исходных текстов, определенная в п. 6.3.

2 Проведено тестирование эталонной реализации. При тестировании использовались две другие независимые реализации. Использовались тесты, определенные в п. 6.2, а также тестовые примеры СТБ 34.101.31.

Тестированию подлежат криптографические алгоритмы, реализованные в программе и определенные в СТБ 34.101.31, включая:

– алгоритмы шифрования в режиме простой замены (см. п. 6.2.1);
– алгоритмы шифрования в режиме сцепления блоков (см. п. 6.2.2);
– алгоритмы шифрования в режиме гаммирования с обратной связью (см. п. 6.2.3);
– алгоритмы шифрования в режиме счетчика (см. п. 6.2.4);
– алгоритм выработки имитовставки (см. п. 6.2.5);
– алгоритмы аутентифицированного шифрования данных (см. п. 6.2.6);
– алгоритмы аутентифицированного шифрования ключа (см. п. 6.2.7);
– алгоритм хэширования (см. п. 6.2.8);
– алгоритмы дискового шифрования (см. п. 6.2.9);
– алгоритмы шифрования с сохранением формата (см. п. 6.2.10);
– алгоритм расширения ключа (см. п. 6.2.11);
– алгоритм преобразования ключа (см. п. 6.2.12).

Если программа не реализует некоторые из алгоритмов, определенных в СТБ 34.101.31, то тесты для них не выполняются.

Для организации тестирования в исходные тексты программы допускается вносить изменения и дополнения, касающиеся:

– способа чтения входных данных;
– способа записи выходных данных.

При внесении модификаций в исходные тексты должен быть проведен анализ корректности внесенных изменений.

При успешном выполнении тест возвращает признак `УСПЕХ`, иначе — `ОШИБКА`. Если при тестировании программы для некоторых входных значений получены результаты отличные от истинных значений, то эксперт по тестированию должен указать эти входные значения программы и результат ее работы, а также, по требованию, результаты промежуточных вычислений экспертам по анализу исходных текстов.

5.4 Анализ исходных текстов

Эксперт проводит анализ исходных текстов путем проверки корректности реализации в испытуемой программе криптографических алгоритмов СТБ 34.101.31. Такой анализ состоит в получении экспертных заключений, касающихся:

- корректности использования локальных переменных (см. п. 6.3.1);
- корректности использования глобальных переменных (см. п. 6.3.2);
- корректности использования констант (см. п. 6.3.3);
- корректности программной логики функций программы (см. п. 6.3.4);
- корректности вызова стандартных функций (см. п. 6.3.5);
- корректности вызова функций программы (см. п. 6.3.6);
- корректности обработки исключительных ситуаций (см. п. 6.3.7);
- корректности реализации криптографических примитивов (см. п. 6.3.8);
- корректности реализации криптографических алгоритмов (см. п. 6.3.9);
- корректности управления секретными данными (см. п. 6.3.10);
- отсутствия недокументированных возможностей (см. п. 6.3.11).

6 Методы испытаний

6.1 Анализ документации

6.1.1 Документ «Спецификация»

При анализе документа «Спецификация» эксперт проверяет, что в нем указаны компоненты и документация, представляемые на испытания.

Если документ «Спецификация» разработан в соответствии с требованиями ЕСПД, то эксперт проверяет, что содержание и оформление документа соответствует ГОСТ 19.202.

6.1.2 Документ «Текст программы»

При анализе документа «Текст программы» эксперт проверяет, что исходные тексты программы, реализующие определенные в СТБ 34.101.31 криптографические алгоритмы, представлены полностью и в виде, который использовался при сборке программы.

Если документ «Текст программы» разработан в соответствии с требованиями ЕСПД, то эксперт проверяет, что содержание и оформление документа соответствует ГОСТ 19.401.

6.1.3 Документ «Описание программы»

При анализе документа «Описание программы» эксперт проверяет выполнение следующих требований:

– в документе должна быть указана информация, однозначно идентифицирующая вызываемые стандартные функции (версия компилятора, используемые стандартные библиотеки и т.п.);

– документ должен определять программные модули, реализующие определенные в СТБ 34.101.31 криптографические алгоритмы;

– описание программы в терминах программных модулей должно соответствовать исходным текстам программы.

Если документ «Описание программы» разработан в соответствии с требованиями ЕСПД, то эксперт проверяет, что содержание и оформление документа соответствует ГОСТ 19.402.

6.1.4 Документ «Руководство программиста»

При анализе документа «Руководство программиста» эксперт проверяет выполнение следующих требований:

– документ должен содержать описание всех доступных для вызова функций, реализующих определенные в СТБ 34.101.31 криптографические алгоритмы;

– описание функций, реализующих определенные в СТБ 34.101.31 криптографические алгоритмы, и условия их использования должны соответствовать исходным текстам программы.

При описании в документации функций должны выполняться следующие условия:

– каждая функция должна иметь описание назначения;

– каждый параметр функции должен иметь описание назначения, типа и, при необходимости, диапазона допустимых значений;

– каждая функция должна иметь описание возвращаемого результата с указанием типа;

– каждая функция должна иметь описание условий, при выполнении которых в ходе работы функции могут возникать ошибочные ситуации, требующие специальной обработки;

– в случае если при реализации криптографического алгоритма используется более одной доступной для вызова функции, должны быть указаны порядок и условия вызова данных функций.

Если документ «Руководство программиста» разработан в соответствии с требованиями ЕСПД, то эксперт проверяет, что содержание и оформление документа соответствует ГОСТ 19.504.

6.2 Тестирование

6.2.1 Алгоритмы шифрования в режиме простой замены

При тестировании реализации алгоритмов шифрования в режиме простой замены выполняются тесты BELT.ECB.1 – BELT.ECB.5.

Входными данными тестов являются ключ $K \in \{0,1\}^{256}$ и сообщение $X \in \{0,1\}^*$.

В тестах для хранения результата зашифрования X на K используются слова $Y, Y' \in \{0,1\}^{|X|}$, а для хранения результата расшифрования Y на K — слово $X' \in \{0,1\}^{|Y|}$.

Тест BELT.ECB.1

1 Задать ключ:

$$K \leftarrow \texttt{2033394D 6C320D09 65201A16 6E62001D}\\ \texttt{67794106 74740E13 6865160D 3D730C11}_{16}.$$

2 Задать сообщение длины 16 октета:

$$X \leftarrow \texttt{00000000 00000000 00000000 00000000}_{16}.$$

3 Испытуемой реализацией выполнить зашифрование X на K и сохранить результат в Y.

4 Если

$$Y = \texttt{D097E3AF 21DC4B88 91688A84 E9A05C51}_{16},$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Тест BELT.ECB.2

1 Задать ключ:

$$K \leftarrow \texttt{348724A4 C1A67667 153DDE59 33884250}$$
$$\texttt{E3248C65 7D413B8C E01C8C9A ADEDF5B9}_{16}.$$

2 Задать сообщение длины 54 октета:

$$X \leftarrow \texttt{46696674 7920666F 75722062 79746520}$$
$$\texttt{6F722066 6F757220 68756E64 72656420}$$
$$\texttt{74686972 74792074 776F2062 6974206D}$$
$$\texttt{65737361 6765}_{16}.$$

3 Испытуемой реализацией выполнить зашифрование X на K и сохранить результат в Y.

4 Если

$$Y = \texttt{6F82F40D 256CE13D EA929981 92F49AAA}$$
$$\texttt{52118F3A 24F588C5 68153EB4 96AE5DDC}$$
$$\texttt{696D8920 D70C3656 02B5731D BB2218F4}$$
$$\texttt{0ED419CF 2288}_{16},$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Примечание – В тесте BELT.ECB.2 используется ключ, который является результатом расширения короткого ключа длины 192 (см. тест BELT.KEX.2 из п. 6.2.11).

Тест BELT.ECB.3

1 Задать ключ:

$$K \leftarrow \texttt{E9DEE72C 8F0C0FA6 2DDB49F4 6F739647}$$
$$\texttt{E9DEE72C 8F0C0FA6 2DDB49F4 6F739647}_{16}.$$

2 Задать сообщение длины 32 октета:

$$X \leftarrow \texttt{54686972 74792074 776F2062 79746520}$$
$$\texttt{6D657373 61676520 28746573 74203129}_{16}.$$

3 Испытуемой реализацией выполнить зашифрование X на K и сохранить результат в Y.

4 Если

$$Y = \texttt{5BD90369 35C95FB3 419F6C59 30CFCE86}$$
$$\texttt{5E52BAC8 CA0E2E3E 2A3267B8 7415C9A3}_{16},$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Примечание – В тесте BELT.ECB.3 используется ключ, который является результатом расширения короткого ключа длины 128 (см. тест BELT.KEX.3 из п. 6.2.11).

Тест BELT.ECB.4

1 Для $i = 1, 2, \ldots, 10000$ выполнить:
1) псевдослучайным методом сгенерировать ключ K;
2) псевдослучайным методом сгенерировать сообщение X длины 2048 октета;
3) испытуемой реализацией выполнить зашифрование X на K и сохранить результат в Y;
4) испытуемой реализацией выполнить расшифрование Y на K и сохранить результат в X';
5) если $X \neq X'$, то возвратить `ОШИБКА`.
2 Возвратить `УСПЕХ`.

Тест BELT.ECB.5

1 Для $i = 1, 2, \ldots, 10000$ выполнить:
1) псевдослучайным методом сгенерировать ключ K;
2) псевдослучайным методом сгенерировать сообщение X длины 2048 октета;
3) испытуемой реализацией выполнить зашифрование X на K и сохранить результат в Y;
4) эталонной реализацией выполнить зашифрование X на K и сохранить результат в Y';
5) если $Y \neq Y'$, то возвратить `ОШИБКА`.
2 Возвратить `УСПЕХ`.

6.2.2 Алгоритмы шифрования в режиме сцепления блоков

При тестировании реализации алгоритмов шифрования в режиме сцепления блоков выполняются тесты BELT.CBC.1 – BELT.CBC.5.

Входными данными тестов являются ключ $K \in \{0,1\}^{256}$, синхропосылка $S \in \{0,1\}^{128}$ и сообщение $X \in \{0,1\}^*$.

В тестах для хранения результата зашифрования X на K и S используются слова $Y, Y' \in \{0,1\}^{|X|}$, а для хранения результата расшифрования Y на K и S — слово $X' \in \{0,1\}^{|Y|}$.

Тест BELT.CBC.1

1 Задать ключ:

$$K \leftarrow \texttt{2033394D 6C320D09 65201A16 6E62001D}$$
$$\texttt{67794106 74740E13 6865160D 3D730C11}_{16}.$$

2 Задать синхропосылку:

$$S \leftarrow \texttt{919ADA90 67B9279E B514BEA1 3F8F2CA8}_{16}.$$

3 Задать сообщение длины 32 октета:

$$X \leftarrow \texttt{54686972 74792074 776F2062 79746520}$$
$$\texttt{6D657373 61676520 28746573 74203129}_{16}.$$

4 Испытуемой реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y.

5 Если

$$Y = \begin{array}{l}\texttt{15168079 B82A03A8 2E059C73 16BD4FCC}\\ \texttt{D349D9FB 619094BF 59D6F7F2 E57B3A7B}_{16},\end{array}$$

то возвратить УСПЕХ, иначе — ОШИБКА.

Тест BELT.CBC.2

1 Задать ключ:

$$K \leftarrow \begin{array}{l}\texttt{348724A4 C1A67667 153DDE59 33884250}\\ \texttt{E3248C65 7D413B8C E01C8C9A ADEDF5B9}_{16}.\end{array}$$

2 Задать синхропосылку:

$$S \leftarrow \texttt{9DEADEC2 621747A6 2A80A7C3 FFA8E347}_{16}.$$

3 Задать сообщение длины 54 октета:

$$X \leftarrow \begin{array}{l}\texttt{46696674 7920666F 75722062 79746520}\\ \texttt{6F722066 6F757220 68756E64 72656420}\\ \texttt{74686972 74792074 776F2062 6974206D}\\ \texttt{65737361 6765}_{16}.\end{array}$$

4 Испытуемой реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y.

5 Если

$$Y = \begin{array}{l}\texttt{104C00E1 E3DCF5A8 C26B614C 4CED22B2}\\ \texttt{8DEE3443 F8BFE217 994ABE89 42B078C2}\\ \texttt{0F940ED6 5C14F365 CCD4810B 3C0A1824}\\ \texttt{868B7F64 C6EC}_{16},\end{array}$$

то возвратить УСПЕХ, иначе — ОШИБКА.

Примечание – В тесте BELT.CBC.2 используется ключ, который является результатом расширения короткого ключа длины 192 (см. тест BELT.KEX.2 из п. 6.2.11).

Тест BELT.CBC.3

1 Задать ключ:

$$K \leftarrow \begin{array}{l}\texttt{E9DEE72C 8F0C0FA6 2DDB49F4 6F739647}\\ \texttt{E9DEE72C 8F0C0FA6 2DDB49F4 6F739647}_{16}.\end{array}$$

2 Задать синхропосылку:

$$S \leftarrow \texttt{D097E3AF 21DC4B88 91688A84 E9A05C51}_{16}.$$

3 Задать сообщение длины 64 октета:

$$X \leftarrow \begin{array}{l}\texttt{00000000 00000000 00000000 00000000}\\ \texttt{00000000 00000000 00000000 00000000}\\ \texttt{00000000 00000000 00000000 00000000}\\ \texttt{00000000 00000000 00000000 00000000}_{16}.\end{array}$$

4 Испытуемой реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y.

5 Если

$$Y = \begin{array}{l}\texttt{AEA07457 FF0DA115 F8C7428F B7EC8614}\\ \texttt{F3F5D44F 1FC8DE48 160E27F0 2A587F44}\\ \texttt{0B83BE7B A4B2AE0B 70262968 D781BE6C}\\ \texttt{9ADDA983 68DD812D C261334A 55AC6844}_{16},\end{array}$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Примечание – В тесте BELT.CBC.3 используется ключ, который является результатом расширения короткого ключа длины 128 (см. тест BELT.KEX.3 из п. 6.2.11).

Тест BELT.CBC.4

1 Для $i = 1, 2, \ldots, 10000$ выполнить:

1) псевдослучайным методом сгенерировать ключ K;

2) псевдослучайным методом сгенерировать синхропосылку S;

3) псевдослучайным методом сгенерировать сообщение X длины 2048 октета;

4) испытуемой реализацией выполнить зашифрование X на K с синхропосылкой S и сохранить результат в Y;

5) испытуемой реализацией выполнить расшифрование Y на K с синхропосылкой S и сохранить результат в X';

6) если $X \neq X'$, то возвратить `ОШИБКА`.

2 Возвратить `УСПЕХ`.

Тест BELT.CBC.5

1 Для $i = 1, 2, \ldots, 10000$ выполнить:

1) псевдослучайным методом сгенерировать ключ K;

2) псевдослучайным методом сгенерировать синхропосылку S;

3) псевдослучайным методом сгенерировать сообщение X длины 2048 октета;

4) испытуемой реализацией выполнить зашифрование X на ключе K с использованием S и сохранить результат в Y;

5) эталонной реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y';

6) если $Y \neq Y'$, то возвратить `ОШИБКА`.

2 Возвратить `УСПЕХ`.

6.2.3 Алгоритмы шифрования в режиме гаммирования с обратной связью

При тестировании реализации алгоритмов шифрования в режиме гаммирования с обратной связью выполняются тесты BELT.CFB.1 – BELT.CFB.5.

Входными данными тестов являются ключ $K \in \{0,1\}^{256}$, синхропосылка $S \in \{0,1\}^{128}$ и сообщение $X \in \{0,1\}^*$.

В тестах для хранения результата зашифрования X на K и S используются слова $Y, Y' \in \{0,1\}^{|X|}$, а для хранения результата расшифрования Y на K и S — слово $X' \in \{0,1\}^{|Y|}$.

Тест BELT.CFB.1

1 Задать ключ:

$$K \leftarrow \texttt{2033394D 6C320D09 65201A16 6E62001D}$$
$$\texttt{67794106 74740E13 6865160D 3D730C11}_{16}.$$

2 Задать синхропосылку:

$$S \leftarrow \texttt{00000000 00000000 00000000 00000000}_{16}.$$

3 Задать сообщение длины 64 октета:

$$X \leftarrow \begin{array}{l}\texttt{D097E3AF 21DC4B88 91688A84 E9A05C51}\\ \texttt{D097E3AF 21DC4B88 91688A84 E9A05C51}\\ \texttt{D097E3AF 21DC4B88 91688A84 E9A05C51}\\ \texttt{D097E3AF 21DC4B88 91688A84 E9A05C51}_{16}.\end{array}$$

4 Испытуемой реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y.

5 Если

$$Y = \begin{array}{l}\texttt{00000000 00000000 00000000 00000000}\\ \texttt{00000000 00000000 00000000 00000000}\\ \texttt{00000000 00000000 00000000 00000000}\\ \texttt{00000000 00000000 00000000 00000000}_{16},\end{array}$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Примечание – В качестве исходного сообщения X используется четыре одинаковых блока, каждый из которых является результатом зашифрования в режиме простой замены синхропосылки S на ключе K (см. тест BELT.ECB.1 из п. 6.2.1). В соответствии с алгоритмом зашифрования в режиме гаммирования с обратной связью результатом точечного теста должно быть сообщение из четырех нулевых блоков.

Тест BELT.CFB.2

1 Задать ключ:

$$K \leftarrow \texttt{348724A4 C1A67667 153DDE59 33884250}$$
$$\texttt{E3248C65 7D413B8C E01C8C9A ADEDF5B9}_{16}.$$

2 Задать синхропосылку:

$$S \leftarrow \texttt{9DEADEC2 621747A6 2A80A7C3 FFA8E347}_{16}.$$

3 Задать сообщение длины 54 октета:

$$X \leftarrow \begin{array}{l}\texttt{46696674 7920666F 75722062 79746520}\\ \texttt{6F722066 6F757220 68756E64 72656420}\\ \texttt{74686972 74792074 776F2062 6974206D}\\ \texttt{65737361 6765}_{16}.\end{array}$$

4 Испытуемой реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y.

5 Если

$$Y = \begin{array}{l}\texttt{E9E2222A C2355263 744D11B2 18C57583}\\ \texttt{63E28168 D79D375D E0F3E276 84AF3D4E}\\ \texttt{6D69E9E5 E7A32793 3C9BFB48 A41B04DD}\\ \texttt{9C41BBBC 9E8D}_{16},\end{array}$$

то возвратить УСПЕХ, иначе — ОШИБКА.

Примечание – В тесте BELT.CFB.2 используется ключ, который является результатом расширения короткого ключа длины 192 (см. тест BELT.KEX.2 из п. 6.2.11).

Тест BELT.CFB.3

1 Задать ключ:

$$K \leftarrow \begin{array}{l}\texttt{E9DEE72C 8F0C0FA6 2DDB49F4 6F739647}\\ \texttt{E9DEE72C 8F0C0FA6 2DDB49F4 6F739647}_{16}.\end{array}$$

2 Задать синхропосылку:

$$S \leftarrow \texttt{D097E3AF 21DC4B88 91688A84 E9A05C51}_{16}.$$

3 Задать сообщение длины 32 октета:

$$X \leftarrow \begin{array}{l}\texttt{54686972 74792074 776F2062 79746520}\\ \texttt{6D657373 61676520 28746573 74203129}_{16}.\end{array}$$

4 Испытуемой реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y.

5 Если

$$Y = \begin{array}{l}\texttt{FAC81D25 8B748161 8FA862ED CE98E334}\\ \texttt{CFBE9553 3542298B 8E61DBC8 816F7175}_{16},\end{array}$$

то возвратить УСПЕХ, иначе — ОШИБКА.

Примечание – В тесте BELT.CFB.3 используется ключ, который является результатом расширения короткого ключа длины 128 (см. тест BELT.KEX.3 из п. 6.2.11).

Тест BELT.CFB.4

1 Для $i = 1, 2, \ldots, 10000$ выполнить:

1) псевдослучайным методом сгенерировать ключ K;

2) псевдослучайным методом сгенерировать синхропосылку S;

3) псевдослучайным методом сгенерировать сообщение X длины 2048 октета;

4) испытуемой реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y;

5) испытуемой реализацией выполнить расшифрование Y на K с использованием S и сохранить результат в X;

6) если $X \neq X'$, то возвратить ОШИБКА.

2 Возвратить УСПЕХ.

Тест BELT.CFB.5

1 Для $i = 1, 2, \ldots, 10000$ выполнить:
1) псевдослучайным методом сгенерировать ключ K;
2) псевдослучайным методом сгенерировать синхропосылку S;
3) псевдослучайным методом сгенерировать сообщение X длины 2048 октета;
4) испытуемой реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y;
5) эталонной реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y';
6) если $Y \neq Y'$, то возвратить `ОШИБКА`.
2 Возвратить `УСПЕХ`.

6.2.4 Алгоритмы шифрования в режиме счетчика

При тестировании реализации алгоритмов шифрования в режиме счетчика выполняются тесты BELT.CTR.1 – BELT.CTR.5.

Входными данными тестов являются ключ $K \in \{0,1\}^{256}$, синхропосылка $S \in \{0,1\}^{128}$ и сообщение $X \in \{0,1\}^*$.

В тестах для хранения результата зашифрования X на K и S используются слова $Y, Y' \in \{0,1\}^{|X|}$, а для хранения результата расшифрования Y на K и S — слово $X' \in \{0,1\}^{|Y|}$.

Тест BELT.CTR.1

1 Задать ключ:

$$K \leftarrow \begin{array}{l}\texttt{2033394D 6C320D09 65201A16 6E62001D}\\ \texttt{67794106 74740E13 6865160D 3D730C11}_{16}.\end{array}$$

2 Задать синхропосылку:

$$S \leftarrow \texttt{919ADA90 67B9279E B514BEA1 3F8F2CA8}_{16}.$$

3 Задать сообщение X, составленное из 64 октетов $\texttt{00}_{16}$.
4 Испытуемой реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y.
5 Если

$$Y = \begin{array}{l}\texttt{E136D829 BCAD5CDE 243A8B02 C3D4ABF6}\\ \texttt{D4F21071 75BD42C0 D7BEED25 B10AD3FA}\\ \texttt{40004EE2 206EC3F8 A986F2AE 9C0C3BD7}\\ \texttt{D097E3AF 21DC4B88 91688A84 E9A05C51}_{16},\end{array}$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Примечание – Так как исходное сообщения X состоит из четырех нулевых блоков, то результатом зашифрования будет четыре блока гаммы шифра. Синхропосылка S и ключ K выбраны таким образом, что для выработки четвертого блока гаммы используется значение накопителя s, состоящее из 16 нулевых октетов. Таким образом, значение четвертого блока гаммы должно быть равно $\texttt{D097E3AF 21DC4B88 91688A84 E9A05C51}_{16}$, что является резуль-

татом зашифрования в режиме простой замены указанного заполнения S на ключе K (см. тест BELT.ECB.1 из п. 6.2.1).

Тест BELT.CTR.2

1 Задать ключ:

$$K \leftarrow \texttt{348724A4 C1A67667 153DDE59 33884250}$$
$$\texttt{E3248C65 7D413B8C E01C8C9A ADEDF5B9}_{16}.$$

2 Задать синхропосылку:

$$S \leftarrow \texttt{9DEADEC2 621747A6 2A80A7C3 FFA8E347}_{16}.$$

3 Задать сообщение длины 54 октета:

$$X \leftarrow \begin{array}{l}\texttt{46696674 7920666F 75722062 79746520}\\ \texttt{6F722066 6F757220 68756E64 72656420}\\ \texttt{74686972 74792074 776F2062 6974206D}\\ \texttt{65737361 6765}_{16}.\end{array}$$

4 Испытуемой реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y.

5 Если

$$Y = \begin{array}{l}\texttt{99EF9A66 19CA4F20 763C6F68 66C1ABE4}\\ \texttt{ED5BD288 655E2970 EFFA2C7F 04085B5E}\\ \texttt{B11185C2 198B6C7E 48FD037C CD0FDA76}\\ \texttt{93388265 ECD3}_{16},\end{array}$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Примечание – В тесте BELT.CTR.2 используется ключ, который является результатом расширения короткого ключа длины 192 (см. тест BELT.KEX.2 из п. 6.2.11).

Тест BELT.CTR.3

1 Задать ключ:

$$K \leftarrow \texttt{E9DEE72C 8F0C0FA6 2DDB49F4 6F739647}$$
$$\texttt{E9DEE72C 8F0C0FA6 2DDB49F4 6F739647}_{16}.$$

2 Задать синхропосылку:

$$S \leftarrow \texttt{D097E3AF 21DC4B88 91688A84 E9A05C51}_{16}.$$

3 Задать сообщение длины 16 октета:

$$X \leftarrow \texttt{B194BAC8 0A08F53B 366D008E 584A5DE4}_{16}.$$

4 Испытуемой реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y.

5 Если

$$Y = \texttt{F6D95919 FF7FFEFB E57604FD 0BF54A5F}_{16},$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Примечание – В тесте BELT.CTR.3 используется ключ, который является результатом расширения короткого ключа длины 128 (см. тест BELT.KEX.3 из п. 6.2.11).

Тест BELT.CTR.4

1 Для $i = 1, 2, \ldots, 10000$ выполнить:
1) псевдослучайным методом сгенерировать ключ K;
2) псевдослучайным методом сгенерировать синхропосылку S;
3) псевдослучайным методом сгенерировать сообщение X длины 2048 октета;
4) испытуемой реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y;
5) испытуемой реализацией выполнить расшифрование Y на K с сипользованием S и сохранить результат в X';
6) если $X \neq X'$, то возвратить `ОШИБКА`.
2 Возвратить `УСПЕХ`.

Тест BELT.CTR.5

1 Для $i = 1, 2, \ldots, 10000$ выполнить:
1) псевдослучайным методом сгенерировать ключ K;
2) псевдослучайным методом сгенерировать синхропосылку S;
3) псевдослучайным методом сгенерировать сообщение X длины 2048 октета;
4) испытуемой реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y;
5) эталонной реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y';
6) если $Y \neq Y'$, то возвратить `ОШИБКА`.
2 Возвратить `УСПЕХ`.

6.2.5 Алгоритм выработки имитовставки

При тестировании реализации алгоритма выработки имитовставки выполняется последовательность тестов BELT.MAC.1 – BELT.MAC.4.

Входными данными тестов являются ключ $K \in \{0,1\}^{256}$ и сообщение $X \in \{0,1\}^*$.

В тестах для хранения имитовставки X на K используются слова $T, T' \in \{0,1\}^{64}$.

Тест BELT.MAC.1

1 Задать ключ:

$$K \leftarrow \texttt{2033394D 6C320D09 65201A16 6E62001D}\ \texttt{67794106 74740E13 6865160D 3D730C11}_{16}.$$

2 Задать сообщение длины 12 октета:

$$X \leftarrow \texttt{B194BAC8 0A08F53B 366D008E}_{16}.$$

3 Испытуемой реализацией выработать имитовставку X на K и сохранить результат в T.
4 Если

$$T = \texttt{21C07374 2A6C6556}_{16},$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Тест BELT.MAC.2

1 Задать ключ:

$$K \leftarrow \texttt{348724A4 C1A67667 153DDE59 33884250}$$
$$\texttt{E3248C65 7D413B8C E01C8C9A ADEDF5B9}_{16}.$$

2 Задать сообщение длины 32 октета:

$$X \leftarrow \texttt{54686972 74792074 776F2062 79746520}$$
$$\texttt{6D657373 61676520 28746573 74203129}_{16}.$$

3 Испытуемой реализацией выработать имитовставку X на K и сохранить результат в T.

4 Если

$$T = \texttt{05B2945C 77D5F6DD}_{16},$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Примечание – В тесте BELT.MAC.2 используется ключ, который является результатом расширения короткого ключа длины 192 (см. тест BELT.KEX.2 из п. 6.2.11).

Тест BELT.MAC.3

1 Задать ключ:

$$K \leftarrow \texttt{E9DEE72C 8F0C0FA6 2DDB49F4 6F739647}$$
$$\texttt{E9DEE72C 8F0C0FA6 2DDB49F4 6F739647}_{16}.$$

2 Задать сообщение длины 54 октета:

$$X \leftarrow \texttt{46696674 7920666F 75722062 79746520}$$
$$\texttt{6F722066 6F757220 68756E64 72656420}$$
$$\texttt{74686972 74792074 776F2062 6974206D}$$
$$\texttt{65737361 6765}_{16}.$$

3 Испытуемой реализацией выработать имитовставку X на K и сохранить результат в T.

4 Если

$$T = \texttt{0B494781 CAE6696E}_{16},$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Примечание – В тесте BELT.MAC.3 используется ключ, который является результатом расширения короткого ключа длины 128 (см. тест BELT.KEX.3 из п. 6.2.11).

Тест BELT.MAC.4

1 Для $i = 1, 2, \ldots, 10000$ выполнить:

1) псевдослучайным методом сгенерировать сообщение X длины 2048 октета;

2) псевдослучайным методом сгенерировать ключ K;

3) испытуемой реализацией выработать имитовставку X на K и сохранить результат в T;
4) эталонной реализацией выработать имитовставку X на K и сохранить результат в T';
5) если $Y \neq Y'$, то возвратить `ОШИБКА`.
2 Возвратить `УСПЕХ`.

6.2.6 Алгоритмы аутентифицируемого шифрования данных

При тестировании реализации алгоритмов аутентифицируемого шифрования данных для схемы DWP (схемы 1) выполняются тесты BELT.DWP.1 – BELT.DWP.5, а при тестировании реализации алгоритмов аутентифицируемого шифрования данных для схемы CHE (схемы 2) — тесты BELT.CHE.1 – BELT.CHE.5

Входными данными тестов являются ключ $K \in \{0,1\}^{256}$, синхропосылка $S \in \{0,1\}^{128}$, сообщение $X \in \{0,1\}^*$ и ассоциированные данные $I \in \{0,1\}^*$.

В тестах для хранения результата зашифрования X на K и Sеге используются слова $Y, Y' \in \{0,1\}^{|X|}$, а для хранения имитовставки пары (X, I) на K и S — слова $T, T' \in \{0,1\}^{64}$. Дополнительно, для хранения результата расшифрования Y на K и S используется слово $X' \in \{0,1\}^{|Y|}$.

Тест BELT.DWP.1

1 Задать ключ:

$$K \leftarrow \texttt{2033394D 6C320D09 65201A16 6E62001D}\ \texttt{67794106 74740E13 6865160D 3D730C11}_{16}.$$

2 Задать синхропосылку:

$$S \leftarrow \texttt{919ADA90 67B9279E B514BEA1 3F8F2CA8}_{16}.$$

3 Задать сообщение длины 32 октета:

$$X \leftarrow \texttt{54686972 74792074 776F2062 79746520}\ \texttt{6D657373 61676520 28746573 74203129}_{16}.$$

4 Задать ассоциированные данные длины 54 октета:

$$I \leftarrow \texttt{46696674 7920666F 75722062 79746520}\ \texttt{6F722066 6F757220 68756E64 72656420}\ \texttt{74686972 74792074 776F2062 6974206D}\ \texttt{65737361 6765}_{16}.$$

5 Испытуемой реализацией установить защиту и сохранить результат в (Y, T).
6 Если

$$Y = \texttt{B55EB15B C8D47CAA 5355AB60 BAA0CED6}\ \texttt{B9976302 14DA27E0 FFCA8856 C52AE2D3}_{16},$$

$$T = \texttt{8F633F70 6F664816}_{16},$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Тест BELT.DWP.2

1 Задать ключ:

$$K \leftarrow \texttt{2033394D 6C320D09 65201A16 6E62001D}$$
$$\texttt{67794106 74740E13 6865160D 3D730C11}_{16}.$$

2 Задать синхропосылку:

$$S \leftarrow \texttt{9DEADEC2 621747A6 2A80A7C3 FFA8E347}_{16}.$$

3 Задать сообщение длины 54 октета:

$$X \leftarrow \texttt{46696674 7920666F 75722062 79746520}$$
$$\texttt{6F722066 6F757220 68756E64 72656420}$$
$$\texttt{74686972 74792074 776F2062 6974206D}$$
$$\texttt{65737361 6765}_{16}.$$

4 Задать ассоциированные данные длины 32 октета:

$$I \leftarrow \texttt{54686972 74792074 776F2062 79746520}$$
$$\texttt{6D657373 61676520 28746573 74203129}_{16}.$$

5 Испытуемой реализацией установить защиту пары (X, I) на K с использованием S и сохранить результат в (Y, T).

6 Если

$$Y = \texttt{48F0D0C3 8111EBCA 02A0C70E 3CCDE8CF}$$
$$\texttt{712B791B B2DD8B43 79DC1526 FCA9F3C6}$$
$$\texttt{2DB8F1AA A3D91F22 1D673F2C FC6476DF}$$
$$\texttt{C7227181 5073}_{16},$$

$$T = \texttt{1E3891C6 05FC961A}_{16},$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Тест BELT.DWP.3

1 Задать ключ:

$$K \leftarrow \texttt{2033394D 6C320D09 65201A16 6E62001D}$$
$$\texttt{67794106 74740E13 6865160D 3D730C11}_{16}.$$

2 Задать синхропосылку:

$$S \leftarrow \texttt{D097E3AF 21DC4B88 91688A84 E9A05C51}_{16}.$$

3 Задать сообщение длины 16 октета:

$$X \leftarrow \texttt{00000000 00000000 00000000 00000000}_{16}.$$

4 Задать ассоциированные данные длины 12 октета:

$$I \leftarrow \texttt{B194BAC8 0A08F53B 366D008E}_{16}.$$

5 Испытуемой реализацией установить защиту пары (X, I) на K с использованием S и сохранить результат в (Y, T).
6 Если

$$Y = \texttt{EE3EBE39 F6D1F123 A8950D43 5CB28ABC}_{16},$$

$$T = \texttt{A9541D79 49102FE4}_{16},$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Тест BELT.DWP.4

1 Для $i = 1, 2, \ldots, 10000$ выполнить:
1) псевдослучайным методом сгенерировать ключ K;
2) псевдослучайным методом сгенерировать синхропосылку S;
3) псевдослучайным методом сгенерировать сообщение X длины 1024 октета;
4) псевдослучайным методом сгенерировать ассоциированные данные I длины 1024 октета;
5) испытуемой реализацией установить защиту пары (X, I) на K с использованием S и сохранить результат в (Y, T);
6) испытуемой реализацией снять защиту с пары (Y, I) на K с использованием S и T и сохранить результат в X';
7) если алгоритм снятии защиты возвратил $\perp$ или $X \neq X'$, то возвратить `ОШИБКА`.
2 Возвратить `УСПЕХ`.

Тест BELT.DWP.5

1 Для $i = 1, 2, \ldots, 10000$ выполнить:
1) псевдослучайным методом сгенерировать ключ K;
2) псевдослучайным методом сгенерировать синхропосылку S;
3) псевдослучайным методом сгенерировать сообщение X длины 1024 октета;
4) псевдослучайным методом сгенерировать ассоциированные данные I длины 1024 октета;
5) испытуемой реализацией установить защиту пары (X, I) на K с использованием S и сохранить результат в (Y, T);
6) эталонной реализацией установить защиту пары (X, I) на K с использованием S и сохранить результат в (Y', T');
7) если $Y \neq Y'$ или $T \neq T'$, то возвратить `ОШИБКА`.
2 Возвратить `УСПЕХ`.

Тест BELT.CHE.1

1 Задать ключ:

$$K \leftarrow \begin{array}{l}\texttt{2033394D 6C320D09 65201A16 6E62001D}\\ \texttt{67794106 74740E13 6865160D 3D730C11}_{16}.\end{array}$$

2 Задать синхропосылку:

$$S \leftarrow \texttt{919ADA90 67B9279E B514BEA1 3F8F2CA8}_{16}.$$

3 Задать сообщение длины 32 октета:

$$X \leftarrow \texttt{54686972 74792074 776F2062 79746520}$$
$$\texttt{6D657373 61676520 28746573 74203129}_{16}.$$

4 Задать ассоциированные данные длины 54 октета:

$$I \leftarrow \texttt{46696674 7920666F 75722062 79746520}$$
$$\texttt{6F722066 6F757220 68756E64 72656420}$$
$$\texttt{74686972 74792074 776F2062 6974206D}$$
$$\texttt{65737361 6765}_{16}.$$

5 Испытуемой реализацией установить защиту пары (X, I) на K с использованием S и сохранить результат в (Y, T).

6 Если

$$Y = \texttt{7653AC1D 1761F70B 6091A92B D0D5F3CA}$$
$$\texttt{81DF53F6 29EC5363 B6B37988 0897B4C8}_{16},$$
$$T = \texttt{3FDE1D16 0F61F81C}_{16},$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Тест BELT.CHE.2

1 Задать ключ:

$$K \leftarrow \texttt{2033394D 6C320D09 65201A16 6E62001D}$$
$$\texttt{67794106 74740E13 6865160D 3D730C11}_{16}.$$

2 Задать синхропосылку:

$$S \leftarrow \texttt{9DEADEC2 621747A6 2A80A7C3 FFA8E347}_{16}.$$

3 Задать сообщение длины 54 октета:

$$X \leftarrow \texttt{46696674 7920666F 75722062 79746520}$$
$$\texttt{6F722066 6F757220 68756E64 72656420}$$
$$\texttt{74686972 74792074 776F2062 6974206D}$$
$$\texttt{65737361 6765}_{16}.$$

4 Задать ассоциированные данные длины 32 октета:

$$I \leftarrow \texttt{54686972 74792074 776F2062 79746520}$$
$$\texttt{6D657373 61676520 28746573 74203129}_{16}.$$

5 Испытуемой реализацией установить защиту пары (X, I) на K с использованием S и сохранить результат в (Y, T).

6 Если

$$Y = \texttt{1D789ED8 5B3F6C38 00CE6F4F 87B2BD7B}$$
$$\texttt{AEC6C574 B41B05A5 AA1CCC6D E30339A6}$$
$$\texttt{419D8548 C4DFBD07 2C1BFBE8 0B970FF3}$$
$$\texttt{6FF366AD 69B8}_{16},$$
$$T = \texttt{6284FEA2 FA54BD84}_{16},$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Тест BELT.CHE.3

1 Задать ключ:

$$K \leftarrow \texttt{2033394D 6C320D09 65201A16 6E62001D}$$
$$\texttt{67794106 74740E13 6865160D 3D730C11}_{16}.$$

2 Задать синхропосылку:

$$S \leftarrow \texttt{D097E3AF 21DC4B88 91688A84 E9A05C51}_{16}.$$

3 Задать сообщение длины 16 октета:

$$X \leftarrow \texttt{00000000 00000000 00000000 00000000}_{16}.$$

4 Задать ассоциированные данные длины 12 октета:

$$I \leftarrow \texttt{B194BAC8 0A08F53B 366D008E}_{16}.$$

5 Испытуемой реализацией установить защиту пары (X, I) на K с использованием S и сохранить результат в (Y, T).

6 Если

$$Y = \texttt{8DA2FA87 DDB8A868 D7FBD41F D603772A}_{16},$$
$$T = \texttt{87E1E64F F3782318}_{16},$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Тест BELT.CHE.4

1 Для $i = 1, 2, \ldots, 10000$ выполнить:

1) псевдослучайным методом сгенерировать ключ K;
2) псевдослучайным методом сгенерировать синхропосылку S;
3) псевдослучайным методом сгенерировать сообщение X длины 1024 октета;
4) псевдослучайным методом сгенерировать ассоциированные данные I длины 1024 октета;
5) испытуемой реализацией установить защиту пары (X, I) на K с использованием S и сохранить результат в (Y, T);
6) испытуемой реализацией снять защиту с пары (Y, I) на K с использованием S и T и сохранить результат в X';
7) если алгоритм снятии защиты возвратил $\perp$ или $X \neq X'$, то возвратить `ОШИБКА`.

2 Возвратить `УСПЕХ`.

Тест BELT.CHE.5

1 Для $i = 1, 2, \ldots, 10000$ выполнить:

1) псевдослучайным методом сгенерировать ключ K;
2) псевдослучайным методом сгенерировать синхропосылку S;
3) псевдослучайным методом сгенерировать сообщение X длины 1024 октета;

4) псевдослучайным методом сгенерировать ассоциированные данные I длины 1024 октета;
5) испытуемой реализацией установить защиту пары (X, I) на K с использованием S и сохранить результат в (Y, T);
6) эталонной реализацией установить защиту пары (X, I) на K с использованием S и сохранить результат в (Y', T');
7) если $Y \neq Y'$ или $T \neq T'$, то возвратить `ОШИБКА`.
2 Возвратить `УСПЕХ`.

6.2.7 Алгоритмы аутентифицируемого шифрования ключа

При тестировании реализации алгоритмов аутентифицируемого шифрования ключа выполняются тесты BELT.KWP.1 – BELT.KWP.5.

Входными данными тестов являются ключ защиты $K \in \{0,1\}^{256}$, защищаемый ключ $X \in \{0,1\}^{8*}$ и его заголовок $I \in \{0,1\}^{128}$.

В тестах для хранения результата защиты пары (X, I) на K используются слова $Y, Y' \in \{0,1\}^{|X|+128}$, а для хранения результата снятия защиты с пары (Y, I) на K — слово $X' \in \{0,1\}^{|Y|-128}$.

Тест BELT.KWP.1

1 Задать ключ защиты:

$$K \leftarrow \begin{array}{l}\texttt{54686972 74792074 776F2062 79746520}\\ \texttt{6D657373 61676520 28746573 74203129}_{16}.\end{array}$$

2 Задать защищаемый ключ длины 32 октета:

$$X \leftarrow \begin{array}{l}\texttt{2033394D 6C320D09 65201A16 6E62001D}\\ \texttt{67794106 74740E13 6865160D 3D730C11}_{16}.\end{array}$$

3 Задать заголовок защищаемого ключа:

$$I \leftarrow \texttt{919ADA90 67B9279E B514BEA1 3F8F2CA8}_{16}.$$

4 Испытуемой реализацией установить защиту пары (X, I) на K и сохранить результат в Y.
5 Если

$$Y = \begin{array}{l}\texttt{CA33D6EB 42021D8D 9074A3E4 C18822B0}\\ \texttt{99A75B9F CFA43E94 2A2856D2 305FB7BC}\\ \texttt{8041F154 81E8EBAA DFE2D0DE EFC4AE1D}_{16},\end{array}$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Тест BELT.KWP.2

1 Задать ключ защиты:

$$K \leftarrow \begin{array}{l}\texttt{54686972 74792074 776F2062 79746520}\\ \texttt{6D657373 61676520 28746573 74203129}_{16}.\end{array}$$

2 Задать защищаемый ключ длины 24 октета:

$$X \leftarrow \begin{array}{l}\texttt{348724A4 C1A67667 153DDE59 33884250}\\ \texttt{E3248C65 7D413B8C}_{16}.\end{array}$$

3 Задать заголовок защищаемого ключа:

$$I \leftarrow \texttt{9DEADEC2 621747A6 2A80A7C3 FFA8E347}_{16}.$$

4 Испытуемой реализацией установить защиту пары (X, I) на K и сохранить результат в Y.

5 Если

$$Y = \begin{array}{l}\texttt{1E2F3EB7 F45F1521 CC7265C3 A13E461C}\\ \texttt{A639D492 CE98EDFD 943AA81F 779052DF}\\ \texttt{494776B1 0DCB4B39}_{16},\end{array}$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Тест BELT.KWP.3

1 Задать ключ защиты:

$$K \leftarrow \begin{array}{l}\texttt{54686972 74792074 776F2062 79746520}\\ \texttt{6D657373 61676520 28746573 74203129}_{16}.\end{array}$$

2 Задать защищаемый ключ длины 16 октета:

$$X \leftarrow \texttt{E9DEE72C 8F0C0FA6 2DDB49F4 6F739647}_{16}.$$

3 Задать заголовок защищаемого ключа:

$$I \leftarrow \texttt{D097E3AF 21DC4B88 91688A84 E9A05C51}_{16}.$$

4 Испытуемой реализацией установить защиту пары (X, I) на K и сохранить результат в Y.

5 Если

$$Y = \begin{array}{l}\texttt{31B682BC 78CED493 D6412953 4280483C}\\ \texttt{4FE76CCF C3E14E9A 9680A641 B3B68C65}_{16},\end{array}$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Тест BELT.KWP.4

1 Для $i = 1, 2, \ldots, 10000$ выполнить:

1) псевдослучайным методом сгенерировать ключ защиты K;
2) псевдослучайным методом сгенерировать защищаемый ключ X длины 32 октета;
3) псевдослучайным методом сгенерировать заголовок защищаемого ключа I;
4) испытуемой реализацией установить защиту пары (X, I) на K и сохранить результат в Y;
5) испытуемой реализацией снять защиту с пары (Y, I) на K и сохранить результат в X';

6) если алгоритм снятии защиты возвратил $\perp$ или $X \neq X'$, то возвратить `ОШИБКА`.
2 Возвратить `УСПЕХ`.

Тест BELT.KWP.5

1 Для $i = 1, 2, \ldots, 10000$ выполнить:
1) псевдослучайным методом сгенерировать ключ K;
2) псевдослучайным методом сгенерировать защищаемый ключ X длины 32 октета;
3) псевдослучайным методом сгенерировать заголовок защищаемого ключа I;
4) испытуемой реализацией установить защиту пары (X, I) на K и сохранить результат в Y;
5) эталонной реализацией установить защиту пары (X, I) на K и сохранить результат в Y';
6) если $Y \neq Y'$, то возвратить `ОШИБКА`.
2 Возвратить `УСПЕХ`.

6.2.8 Алгоритм хэширования

При тестировании реализации алгоритма хэширования выполняются тесты BELT.HSH.1 – BELT.HSH.5.

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

В тестах для хранения хэш-значения X используются слова $Y, Y' \in \{0,1\}^{256}$.

Тест BELT.HSH.1

1 Задать сообщение длины 12 октета:

$$X \leftarrow \texttt{B194BAC8 0A08F53B 366D008E}_{16}.$$

2 Испытуемой реализацией вычислить хэш-значение X и сохранить результат в Y.
3 Если

$$Y = \begin{array}{l}\texttt{EA4369B3 D0F0A302 B6A5E2A8 6E8255DA}\\ \texttt{4C096573 FBE89AFC EA6A0609 61B6FEA3}_{16},\end{array}$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Тест BELT.HSH.2

1 Задать сообщение длины 32 октета:

$$X \leftarrow \begin{array}{l}\texttt{54686972 74792074 776F2062 79746520}\\ \texttt{6D657373 61676520 28746573 74203129}_{16}.\end{array}$$

2 Испытуемой реализацией вычислить хэш-значение X и сохранить результат в Y.
3 Если

$$Y = \begin{array}{l}\texttt{7F526F9C E5BBC4B1 4A8B9F9B 45E88CE2}\\ \texttt{00CE7ADD 6E4AE62D CBB4EFA8 1EE5FC58}_{16},\end{array}$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Тест BELT.HSH.3

1 Задать сообщение длины 54 октета:

$$X \leftarrow \begin{array}{l}\texttt{46696674 7920666F 75722062 79746520}\\ \texttt{6F722066 6F757220 68756E64 72656420}\\ \texttt{74686972 74792074 776F2062 6974206D}\\ \texttt{65737361 6765}_{16}.\end{array}$$

2 Испытуемой реализацией вычислить хэш-значение X и сохранить результат в Y.

3 Если

$$Y = \begin{array}{l}\texttt{2AB0EF11 60772379 3F9F32D7 FDC66781}\\ \texttt{3BA71863 17BDE4F9 2854F673 C0D04A48}_{16},\end{array}$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Тест BELT.HSH.4

1 Задать сообщение X, состоящее из 1000000 октетов $\texttt{61}_{16}$ (шестнадцатеричное представление символа ‘a’ в коде ASCII).

2 Испытуемой реализацией вычислить хэш-значение X и сохранить результат в Y.

3 Если

$$Y = \begin{array}{l}\texttt{98001732 AC6BD9A3 B03B6688 6320EC8A}\\ \texttt{3E438255 81E10779 130B02FB D67E21E5}_{16},\end{array}$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Тест BELT.HSH.5

1 Для $i = 1, 2, \ldots, 10000$ выполнить:

1) псевдослучайным методом сгенерировать сообщения X длины 2048 октета;
2) испытуемой реализацией вычислить хэш-значение X и сохранить результат в Y;
3) эталонной реализацией вычислить хэш-значение X и сохранить результат в Y';
4) если $Y \neq Y'$, то возвратить `ОШИБКА`.

2 Возвратить `УСПЕХ`.

6.2.9 Алгоритмы дискового шифрования

При тестировании реализации алгоритмов блокового шифрования выполняются тесты BELT.BDE.1 – BELT.BDE.4, а при тестировании реализации алгоритмов секторного шифрования — тесты BELT.SDE.1 – BELT.SDE.4.

Входными данными тестов являются ключ $K \in \{0,1\}^{256}$, синхропосылка $S \in \{0,1\}^{128}$ и сообщение $X \in \{0,1\}^{128*}$.

В тестах для хранения результата зашифрования X на K и S используются слова $Y, Y' \in \{0,1\}^{|X|}$, а для хранения результата расшифрования Y на K и S — слово $X' \in \{0,1\}^{|Y|}$.

Тест BELT.BDE.1

1 Задать ключ:

$$K \leftarrow \texttt{E9DEE72C 8F0C0FA6 2DDB49F4 6F739647}\\ \texttt{06075316 ED247A37 39CBA383 03A98BF6}_{16}.$$

2 Задать синхропосылку:

$$S \leftarrow \texttt{BE329713 43FC9A48 A02A885F 194B09A1}_{16}.$$

3 Задать сообщение длины 48 октета:

$$X \leftarrow \texttt{B194BAC8 0A08F53B 366D008E 584A5DE4}\\ \texttt{8504FA9D 1BB6C7AC 252E72C2 02FDCE0D}\\ \texttt{5BE3D612 17B96181 FE6786AD 716B890B}_{16}.$$

4 Испытуемой реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y.

5 Если

$$Y = \texttt{E9CAB32D 879CC50C 10378EB0 7C10F263}\\ \texttt{07257E2D BE2B854C BC9F3828 2D59D6A7}\\ \texttt{7F952001 C5D1244F 53210A27 C216D4BB}_{16},$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Тест BELT.BDE.2

1 Задать ключ:

$$K \leftarrow \texttt{92BD9B1C E5D14101 5445FBC9 5E4D0EF2}\\ \texttt{682080AA 227D642F 2687F934 90405511}_{16}.$$

2 Задать синхропосылку:

$$S \leftarrow \texttt{7ECDA4D0 1544AF8C A58450BF 66D2E88A}_{16}.$$

3 Задать сообщение длины 48 октета:

$$X \leftarrow \texttt{E12BDC1A E28257EC 703FCCF0 95EE8DF1}\\ \texttt{C1AB7638 9FE678CA F7C6F860 D5BB9C4F}\\ \texttt{F33C657B 637C306A DD4EA779 9EB23D31}_{16}.$$

4 Испытуемой реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y.

5 Если

$$Y = \texttt{7041BC22 6352C706 D00EA8EF 23CFE46A}\\ \texttt{FAE11857 7D037FAC DC36E4EC C1F65746}\\ \texttt{09F23694 3FB809E1 BEE4A1C6 86C13ACC}_{16},$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Тест BELT.BDE.3

1 Для $i = 1, 2, \ldots, 10000$ выполнить:
1) псевдослучайным методом сгенерировать ключ K;
2) псевдослучайным методом сгенерировать синхропосылку S;
3) псевдослучайным методом сгенерировать сообщение X длины 2048 октета;
4) испытуемой реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y;
5) испытуемой реализацией выполнить расшифрование Y на K с использованием S и сохранить результат в X';
2 Тест выполнен успешно, если $X'_i = X_i$ для $i = 1, 2, \ldots, 10000$.

Тест BELT.BDE.4

1 Для $i = 1, 2, \ldots, 10000$ выполнить:
1) псевдослучайным методом сгенерировать ключ K;
2) псевдослучайным методом сгенерировать синхропосылку S;
3) псевдослучайным методом сгенерировать сообщение X длины 2048 октета;
4) испытуемой реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y;
5) эталонной реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y';
6) если $Y \neq Y'$, то возвратить `ОШИБКА`.
2 Возвратить `УСПЕХ`.

Тест BELT.SDE.1

1 Задать ключ:

$$K \leftarrow \texttt{E9DEE72C 8F0C0FA6 2DDB49F4 6F739647}\ \texttt{06075316 ED247A37 39CBA383 03A98BF6}_{16}.$$

2 Задать синхропосылку:

$$S \leftarrow \texttt{BE329713 43FC9A48 A02A885F 194B09A1}_{16}.$$

3 Задать сообщение длины 48 октета:

$$X \leftarrow \texttt{B194BAC8 0A08F53B 366D008E 584A5DE4}\ \texttt{8504FA9D 1BB6C7AC 252E72C2 02FDCE0D}\ \texttt{5BE3D612 17B96181 FE6786AD 716B890B}_{16}.$$

4 Испытуемой реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y.
5 Если

$$Y = \texttt{1FCBB018 52003D60 B66024C5 08608BAA}\ \texttt{2C21AF1E 884CF311 54D3077D 4643CF22}\ \texttt{49EB2F5A 68E4BA01 9D90211A 81D690D9}_{16},$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Тест BELT.SDE.2

1 Задать ключ:

$$K \leftarrow \texttt{92BD9B1C E5D14101 5445FBC9 5E4D0EF2}\ \texttt{682080AA 227D642F 2687F934 90405511}_{16}.$$

2 Задать синхропосылку:

$$S \leftarrow \texttt{7ECDA4D0 1544AF8C A58450BF 66D2E88A}_{16}.$$

3 Задать сообщение длины 48 октета:

$$X \leftarrow \texttt{E12BDC1A E28257EC 703FCCF0 95EE8DF1}\ \texttt{C1AB7638 9FE678CA F7C6F860 D5BB9C4F}\ \texttt{F33C657B 637C306A DD4EA779 9EB23D31}_{16}.$$

4 Испытуемой реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y.

5 Если

$$Y = \texttt{E9FDF3F7 88657332 E6C46FCF 5251B8A6}\ \texttt{D43543A9 3E323383 7DB15711 83A6EF4D}\ \texttt{7FEB5CDF 999E1A3F 51A5A338 1BEB7FA5}_{16},$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Тест BELT.SDE.3

1 Для $i = 1, 2, \ldots, 10000$ выполнить:

1) псевдослучайным методом сгенерировать ключ K;
2) псевдослучайным методом сгенерировать синхропосылку S;
3) псевдослучайным методом сгенерировать сообщение X длины 2048 октета;
4) испытуемой реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y;
5) испытуемой реализацией выполнить расшифрование Y на K с использованием S и сохранить результат в X';
6) если $X \neq X'$, то возвратить `ОШИБКА`.

2 Возвратить `УСПЕХ`.

Тест BELT.SDE.4

1 Для $i = 1, 2, \ldots, 10000$ выполнить:

1) псевдослучайным методом сгенерировать ключ K;
2) псевдослучайным методом сгенерировать синхропосылку S;
3) псевдослучайным методом сгенерировать сообщение X длины 2048 октета;
4) испытуемой реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y;
5) эталонной реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y';
6) если $Y \neq Y'$, то возвратить `ОШИБКА`.

2 Возвратить `УСПЕХ`.

6.2.10 Алгоритмы шифрования с сохранением формата

В ходе тестирования реализации режима счетчика алгоритма шифрования выполняются тесты BELT.FMT.1 —BELT.FMT.5.

Входными данными тестов являются ключ $K \in \{0,1\}^{256}$ синхропосылка $S \in \{0,1\}^{128}$ и слово $X \in \mathbb{Z}_m^n$.

В тестах для хранения результата зашифрования X на K и S используются слова $Y, Y' \in \mathbb{Z}_m^n$, а для хранения результата расшифрования Y на K и S — слово $X' \in \mathbb{Z}_m^n$.

Тест BELT.FMT.1

1 Задать ключ:

$$K \leftarrow \texttt{E9DEE72C 8F0C0FA6 2DDB49F4 6F739647}\\ \texttt{06075316 ED247A37 39CBA383 03A98BF6}_{16}.$$

2 Задать синхропосылку:

$$S \leftarrow \texttt{BE329713 43FC9A48 A02A885F 194B09A1}_{16}.$$

3 Задать слово из алфавита размера $m = 10$ и длины 10 символов:

$$X \leftarrow \texttt{0, 1, 2, 3, 4, 5, 6, 7, 8, 9.}$$

4 Испытуемой реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y.

5 Если

$$Y = \texttt{9, 1, 6, 1, 8, 9, 0, 0, 3, 2,}$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Тест BELT.FMT.2

1 Задать ключ:

$$K \leftarrow \texttt{E9DEE72C 8F0C0FA6 2DDB49F4 6F739647}\\ \texttt{06075316 ED247A37 39CBA383 03A98BF6}_{16}.$$

2 Задать синхропосылку:

$$S \leftarrow \texttt{BE329713 43FC9A48 A02A885F 194B09A1}_{16}.$$

3 Задать слово из алфавита размера $m = 58, n = 21$ и длины 21 символ:

$$X \leftarrow \texttt{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,}\\ \texttt{12, 13, 14, 15, 16, 17, 18, 19, 20.}$$

4 Испытуемой реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y.

5 Если

$$Y = \texttt{54, 57, 12, 33, 7, 45, 52, 13, 36, 7, 7,}\\ \texttt{15, 10, 26, 9, 53, 30, 51, 39, 19, 51,}$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Тест BELT.FMT.3

1 Задать ключ:

$$K \leftarrow \texttt{E9DEE72C 8F0C0FA6 2DDB49F4 6F739647}\\ \texttt{06075316 ED247A37 39CBA383 03A98BF6}_{16}.$$

2 Задать синхропосылку:

$$S \leftarrow \texttt{BE329713 43FC9A48 A02A885F 194B09A1}_{16}.$$

3 Задать слово из алфавита размера $m = 65536, n = 17$ и длины 17 символов:

$$X \leftarrow \texttt{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,}\\ \texttt{12, 13, 14, 15, 16.}$$

4 Испытуемой реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y.

5 Если

$$Y = \texttt{10699, 44372, 28885, 6592, 7111, 60658,}\\ \texttt{33096, 8253, 61778, 315, 19436, 35582,}\\ \texttt{15517, 61117, 59921, 55117, 50041,}$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Тест BELT.FMT.4

1 Для $i = 1, 2, \ldots, 10000$ выполнить:

1) псевдослучайным методом сгенерировать ключ K;

2) псевдослучайным методом сгенерировать синхропосылку S;

3) псевдослучайным методом сгенерировать сообщение X из алфавита размера $m = 10$ и длины 10 символов;

4) Испытуемой реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y.

5) Испытуемой реализацией выполнить расшифрование Y на K с использованием S и сохранить результат в X'.

6) если $X \neq X'$, то возвратить `ОШИБКА`.

2 Возвратить `УСПЕХ`.

Тест BELT.FMT.5

1 Для $i = 1, 2, \ldots, 10000$ выполнить:

1) псевдослучайным методом сгенерировать ключ K;

2) псевдослучайным методом сгенерировать синхропосылку S;

3) псевдослучайным методом сгенерировать слово X из алфавита размера $m = 10$ и длины 10 символов;

4) Испытуемой реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y.

5) эталонной реализацией выполнить зашифрование X на K с использованием S и сохранить результат в Y'.

6) если $Y \neq Y'$, то возвратить `ОШИБКА`.

2 Возвратить `УСПЕХ`.

6.2.11 Алгоритм расширения ключа

При тестировании реализации алгоритма расширения ключа выполняются тесты BELT.KEX.1 —BELT.KEX.3.

Входными данными тестов является ключ $K \in \{0,1\}^{32n}$, где $n \in \{4,6,8\}$.

В тестах для хранения расширения K используется слово $K' \in \{0,1\}^{256}$.

Тест BELT.KEX.1

1 Задать ключ длины 32 октета:

$$K \leftarrow \texttt{2033394D 6C320D09 65201A16 6E62001D}$$
$$\texttt{67794106 74740E13 6865160D 3D730C11}_{16}.$$

2 Испытуемой реализацией выполнить расширение K и сохранить результат в K'.

3 Если

$$K' = \texttt{2033394D 6C320D09 65201A16 6E62001D}$$
$$\texttt{67794106 74740E13 6865160D 3D730C11}_{16},$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Тест BELT.KEX.2

1 Задать ключ длины 24 октета:

$$K \leftarrow \texttt{348724A4 C1A67667 153DDE59 33884250}$$
$$\texttt{E3248C65 7D413B8C}_{16}.$$

2 Испытуемой реализацией выполнить расширение K и сохранить результат в K'.

3 Если

$$K' = \texttt{348724A4 C1A67667 153DDE59 33884250}$$
$$\texttt{E3248C65 7D413B8C E01C8C9A ADEDF5B9}_{16},$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Тест BELT.KEX.3

1 Задать ключ длины 16 октета:

$$K \leftarrow \texttt{E9DEE72C 8F0C0FA6 2DDB49F4 6F739647}_{16}.$$

2 Испытуемой реализацией выполнить расширение K и сохранить результат в K'.

3 Если

$$K' = \texttt{E9DEE72C 8F0C0FA6 2DDB49F4 6F739647}$$
$$\texttt{E9DEE72C 8F0C0FA6 2DDB49F4 6F739647}_{16},$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

6.2.12 Алгоритм преобразования ключа

При тестировании реализации алгоритма преобразования ключа выполняются тесты BELT.KRP.1 – BELT.KRP.6.

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

В тестах для хранения нового ключа, полученного для I и m по X уровня D, используется слово $Y \in \{0,1\}^m$.

Тест BELT.KRP.1

1 Задать преобразуемый ключ длины 32 октета:

$$X \leftarrow \texttt{2033394D 6C320D09 65201A16 6E62001D 67794106 74740E13 6865160D 3D730C11}_{16}.$$

2 Задать уровень преобразуемого ключа:

$$D \leftarrow \texttt{01000000 00000000 00000000}_{16}.$$

3 Задать заголовок нового ключа:

$$I \leftarrow \texttt{919ADA90 67B9279E B514BEA1 3F8F2CA8}_{16}.$$

4 Задать длину нового ключа: $m \leftarrow 256$.

5 Испытуемой реализацией преобразовать X уровня D с использованием I, m и сохранить результат в Y.

6 Если

$$Y = \texttt{39B22F1A EB3BDA3A F2D15C9E F4D7E1B9 E04CBB98 FEE03A03 E521CE4B DA191B38}_{16},$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Тест BELT.KRP.2

1 Задать преобразуемый ключ длины 32 октета:

$$X \leftarrow \texttt{2033394D 6C320D09 65201A16 6E62001D 67794106 74740E13 6865160D 3D730C11}_{16}.$$

2 Задать уровень преобразуемого ключа:

$$D \leftarrow \texttt{01000000 00000000 00000000}_{16}.$$

3 Задать заголовок нового ключа:

$$I \leftarrow \texttt{9DEADEC2 621747A6 2A80A7C3 FFA8E347}_{16}.$$

4 Задать длину нового ключа: $m \leftarrow 192$.

5 Испытуемой реализацией преобразовать X уровня D с использованием I, m и сохранить результат в Y.

6 Если

$$Y = \texttt{59A50D2B EA321BCE 681B21B3 B745345A}$$
$$\texttt{2F642D5A 7C68C4DC}_{16},$$

то возвратить УСПЕХ, иначе — ОШИБКА.

Тест BELT.KRP.3

1 Задать преобразуемый ключ длины 32 октета:

$$X \leftarrow \texttt{2033394D 6C320D09 65201A16 6E62001D}$$
$$\texttt{67794106 74740E13 6865160D 3D730C11}_{16}.$$

2 Задать уровень преобразуемого ключа:

$$D \leftarrow \texttt{01000000 00000000 00000000}_{16}.$$

3 Задать заголовок нового ключа:

$$I \leftarrow \texttt{D097E3AF 21DC4B88 91688A84 E9A05C51}_{16}.$$

4 Задать длину нового ключа: $m \leftarrow 128$.

5 Испытуемой реализацией преобразовать X уровня D с использованием I, m и сохранить результат в Y.

6 Если

$$Y = \texttt{88743329 791EE70C BE6B3438 FEAC2FB4}_{16},$$

то возвратить УСПЕХ, иначе — ОШИБКА.

Тест BELT.KRP.4

1 Задать преобразуемый ключ длины 24 октета:

$$X \leftarrow \texttt{348724A4 C1A67667 153DDE59 33884250}$$
$$\texttt{E3248C65 7D413B8C}_{16}.$$

2 Задать уровень преобразуемого ключа:

$$D \leftarrow \texttt{01000000 00000000 00000000}_{16}.$$

3 Задать заголовок нового ключа:

$$I \leftarrow \texttt{919ADA90 67B9279E B514BEA1 3F8F2CA8}_{16}.$$

4 Задать длину нового ключа: $m \leftarrow 192$.

5 Испытуемой реализацией преобразовать X уровня D с использованием I, m и сохранить результат в Y.

6 Если

$$Y = \texttt{BFCBEAA0 5620BA4F A04AE7CD 482F7B97}$$
$$\texttt{15FBBF63 459B1C66}_{16},$$

то возвратить УСПЕХ, иначе — ОШИБКА.

Тест BELT.KRP.5

1 Задать преобразуемый ключ длины 24 октета:

$$X \leftarrow \texttt{348724A4 C1A67667 153DDE59 33884250 E3248C65 7D413B8C}_{16}.$$

2 Задать уровень преобразуемого ключа:

$$D \leftarrow \texttt{01000000 00000000 00000000}_{16}.$$

3 Задать заголовок нового ключа:

$$I \leftarrow \texttt{9DEADEC2 621747A6 2A80A7C3 FFA8E347}_{16}.$$

4 Задать длину нового ключа: $m \leftarrow 128$.
5 Испытуемой реализацией преобразовать X уровня D с использованием I, m и сохранить результат в Y.
6 Если

$$Y = \texttt{9D8152A7 EE9DDDC9 B42161FE 1FFB8C84}_{16},$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

Тест BELT.KRP.6

1 Задать преобразуемый ключ длины 16 октета:

$$X \leftarrow \texttt{E9DEE72C 8F0C0FA6 2DDB49F4 6F739647}_{16}.$$

2 Задать уровень преобразуемого ключа:

$$D \leftarrow \texttt{01000000 00000000 00000000}_{16}.$$

3 Задать заголовок нового ключа:

$$I \leftarrow \texttt{D097E3AF 21DC4B88 91688A84 E9A05C51}_{16}.$$

4 Задать длину нового ключа: $m \leftarrow 128$.
5 Испытуемой реализацией преобразовать X уровня D с использованием I, m и сохранить результат в Y.
6 Если

$$Y = \texttt{5285AB0A EE08A2A5 4AF51233 2E04C8B2}_{16},$$

то возвратить `УСПЕХ`, иначе — `ОШИБКА`.

6.3 Анализ исходных текстов

6.3.1 Корректность использования локальных переменных

Анализ корректности использования локальных переменных проводится для всех функций программы.

Под функцией понимается часть программы, которая выполняет специфические действия и описывается типом возвращаемого значения, именем функции, формальными параметрами. Выполнение функции осуществляется посредством вызова из программы или другой функции. Данному термину в языках программирования соответствуют такие понятия как «функция», «процедура», «метод» и т.п.

Для каждой локальной переменной v функции f эксперт определяет языковые конструкции f, в которых v встречается, и выполняет следующие проверки:

1 При использовании v в левой части оператора присваивания тип присваиваемого значения должен совпадать с типом v, в противном случае эксперт проверяет корректность результата, учитывая стандартные правила преобразования типов, определенные в используемом языке программирования.

2 Перед использованием значения переменной v должна быть выполнена ее инициализация.

3 Обращение на чтение/запись к переменной v должно происходить в пределах установленных для нее границ, в частности, если v является переменной составного типа, то обращение к элементам v должно происходить в пределах заданных размерностей.

4 Если v является переменной вещественного типа, то ее использование в операциях сравнения запрещено.

5 Если память для v выделяется в динамической области, то перед каждым выходом из f динамическая память должна быть освобождена. После освобождения памяти не должно быть языковых конструкций, ссылающихся на нее.

Примечание — В языках программирования, снабженных средствами «сборки мусора», освобождение динамической памяти, выделяемой для локальной переменной, может быть неявным.

6.3.2 Корректность использования глобальных переменных

Для каждой глобальной переменной v эксперт определяет языковые конструкции программы, в которых v встречается. Далее выполняются проверки 1 – 4 из п. 6.3.1 и следующие проверки:

1 Если память для v выделяется в динамической области, то перед каждым выходом из программы динамическая память должна быть освобождена. После освобождения памяти не должно быть языковых конструкций, ссылающихся на нее.

2 Если v может использоваться в многопоточном режиме работы программы, то должны быть реализованы механизмы, обеспечивающие разграничение доступа к v (механизмы синхронизации доступа к глобальной переменной), при этом данные механизмы не должны блокировать доступ к v на неограниченное время.

Примечание – В языках программирования, снабженных средствами «сборки мусора», освобождение динамической памяти, выделяемой для глобальной переменной, может быть неявным.

6.3.3 Корректность использования констант

Эксперт определяет языковые конструкции программы, в которых встречаются следующие константы:

– значения подстановки H (таблица 2 п. 6.1.3 СТБ 34.101.31);
– значения переменной s (шаг 2 п. 7.5.3 СТБ 34.101.31);
– значения переменной t (шаг 2 п. 7.6.3 и шаг 2 п. 7.6.4 СТБ 34.101.31);

– значения переменной s (шаг 2 п. 7.8.3 СТБ 34.101.31);
– значения переменной h (шаг 2 п. 7.8.3 СТБ 34.101.31);
– значения констант $C_0, \ldots, C_5$ (п. 7.10.4 СТБ 34.101.31);
– значения переменной r (шаг 1 п. 8.2.3 СТБ 34.101.31).

Для каждой языковой конструкции эксперт проверяет, что константы заданы правильно.

6.3.4 Корректность программной логики функций программы

Для каждой функции программы эксперт выполняет следующие проверки:

1 Проверка допустимости переданных параметров и используемых глобальных переменных выполняется до их использования. Проверка может не выполняться, если в документации или в комментариях к функции оговорены ограничения на входные данные, при которых функция работает правильно, и эти ограничения соблюдаются для входных данных во всех вызовах функции.

2 Все заданные варианты условных переходов возможны.

3 Все адреса безусловных переходов доступны.

4 Каждый цикл завершается за конечное число шагов, т.е. завершение цикла гарантировано.

5 После выполнения операторов функции завершение функции гарантировано: достигается одна из точек выхода из функции.

6 Отсутствуют недостижимые участки кода.

7 Цепочки последовательных действий (например, открытие файла, чтение из файла, закрытие файла) корректны. Проверка выполняется, если в функции требуется выполнить некоторое действие, требующее определенной последовательности операций.

6.3.5 Корректность вызова стандартных функций

Эксперт проверяет, что в документации, комментариях исходных текстов программ или конфигурационных файлах указана информация, однозначно идентифицирующая вызываемые стандартные функции (версии компилятора, используемых стандартных библиотек и т.п.).

Для каждого вызова стандартной функции в программе эксперт проверяет:

1 Типы и значения параметров, фактически переданных в функцию, соответствуют типам и допустимым значениям параметров функции, указанным в документации на функцию (с учетом стандартных правил преобразования типов языка программирования).

2 Если в документации на функцию указано, что функция возвращает значение, то проводится анализ корректности использования возвращаемого значения, например, корректность использования в операторе присваивания, допустимость игнорирования возвращаемого значения и т.п.

3 Если в документации на функцию указано, что вызов функции может привести к возникновению исключительной ситуации или ошибки, проверяется наличие и корректность обработки исключительной ситуации.

4 Если в документации на функцию указано, что до и после вызова функции должны выполняться определенные действия, то проверяется наличие и корректность выполнения требуемых действий.

6.3.6 Корректность вызова функций программы

Эксперт проверяет, что в документации или комментариях исходных текстов программ для каждой функции программы указана информация, определяющая:

– допустимые входные параметры и возвращаемые значения функции;

– условия, при выполнении которых в ходе работы функции могут возникать исключительные ситуации (при наличии);

– действия, которые должны выполняться до и(или) после вызова функции (при наличии).

Для каждого вызова функции программы эксперт выполняет следующие проверки:

1 Типы и значения параметров, фактически переданных в функцию, соответствуют типам и допустимым значениям параметров функции (с учетом стандартных правил преобразования типов языка программирования).

2 Если функция возвращает значение, то проводится анализ корректности использования возвращаемого значения, например, корректность использования в операторе присваивания, допустимость игнорирования возвращаемого значения и т.п.

3 Если вызов функции может привести к возникновению исключительной ситуации или ошибки, проверяется наличие и корректность обработки исключительной ситуации.

4 Если до и после вызова функции должны выполняться определенные действия, то проверяется наличие и корректность выполнения требуемых действий.

5 Если функция использует глобальные переменные, то проверяется наличие инициализации данных переменных.

6.3.7 Корректность обработки исключительных ситуаций

Под исключительной ситуацией понимается ошибочная ситуация, возникающая при выполнении программы и требующая специальной обработки. Данному термину в языках программирования соответствует такие понятия как «ошибка», «исключение» и т.п.

Для анализа корректности обработки исключительных ситуаций эксперт формирует список функций, включающий стандартные функции и функции программы, вызов которых может приводить к возникновению исключительной ситуации.

Для каждого вызова функции из составленного списка эксперт проверяет:

1 После каждого вызова функции имеются проверка на случай возникновения исключительной ситуации и соответствующая обработка исключительной ситуации.

2 При проверке и обработке исключительной ситуации учтены все возможные виды исключительных ситуаций, возникновение которых возможно для вызываемой функции.

3 Исключительные ситуации обрабатываются адекватно (возвращаются верные коды ошибок и сообщения об ошибках и т.п.).

6.3.8 Корректность реализации криптографических примитивов

Криптографический примитив — это определенное в СТБ 34.101.31 вспомогательное преобразование, являющееся композиционной частью некоторого криптографического алгоритма.

В СТБ 34.101.31 определены следующие криптографические примитивы:

– преобразования G_r, где $r = 5, 13, 21$ (п. 6.1.3 СТБ 34.101.31);

– алгоритмов `belt-block` и `belt-block`$^{-1}$, (п. 6.1.4, 6.1.5 СТБ 34.101.31);

– алгоритмов `belt-wblock` и `belt-wblock`$^{-1}$, (п. 6.2.4, 6.2.5 СТБ 34.101.31);

– алгоритма `belt-compress` (п. 6.3.3 СТБ 34.101.31);

– преобразований φ_1, φ_2 и отображения ψ (п. 7.5.3 СТБ 34.101.31);

– операции $*$ (п. 4.1 СТБ 34.101.31);

– алгоритма `belt-32block` (п. 7.10.3 СТБ 34.101.31).

Примечание – Операция $*$ задается в СТБ 34.101.31 не как алгоритмическая последовательность шагов, а как отображение «аргументы $\rightarrow$ результат», т.е. функционально. Функционально эквивалентные алгоритмы могут быть устроены по-разному, поэтому требуется оценка соответствия алгоритмического описания функциональному.

Анализируя структуру программы и используя документацию, эксперт формирует список криптографических примитивов, реализованных в программе. Для каждого примитива $g : A \rightarrow B$, осуществляющего отображение множества A в множество B, эксперт проверяет:

– наличие реализации примитива g в виде отдельной функции, части функции или композиции нескольких функций;

– тождественность реализации примитива g спецификации;

– отсутствие в g операций, не используемых для реализации примитива (наличие операций, не предусмотренных спецификацией на примитив, отражается в приложении к протоколу результатов анализа исходных текстов).

Допускается, что действие отображения g определено на множестве A^*, которое является подмножеством A. В этом случае эксперт дополнительно проверяет, что при выполнении программы прообразы отображения g всегда являются элементами A^*.

6.3.9 Корректность реализации криптографических алгоритмов

В СТБ 34.101.31 определены следующие криптографические алгоритмы:

– алгоритмы шифрования в режиме простой замены (п. 7.1 СТБ 34.101.31);

– алгоритмы шифрования в режиме сцепления блоков (п. 7.2 СТБ 34.101.31);

– алгоритмы шифрования в режиме гаммирования с обратной связью (п. 7.3 СТБ 34.101.31);

– алгоритмы шифрования в режиме счетчика (п. 7.4 СТБ 34.101.31);

– алгоритм выработки имитовставки (п. 7.5 СТБ 34.101.31);

– алгоритмы аутентифицируемого шифрования данных (п. 7.6 СТБ 34.101.31);

– алгоритмы аутентифицируемого шифрования ключа (п. 7.7 СТБ 34.101.31);

– алгоритм хэширования (п. 7.8 СТБ 34.101.31);

– алгоритмы дискового шифрования (п. 7.9 СТБ 34.101.31);

– алгоритмы шифрования с сохранением формата (п. 7.10 СТБ 34.101.31);

– алгоритм расширения ключа (п. 8.1 СТБ 34.101.31);

– алгоритм преобразования ключа (п. 8.2 СТБ 34.101.31).

Анализируя структуру программы и используя документацию, эксперт формирует список криптографических алгоритмов, реализованных в программе. Для каждого алгоритма $f : X \times \Theta \rightarrow Y$, который ставит в соответствие входным данным $x \in X$ и параметру $\theta \in \Theta$ результат криптографического преобразования $y \in Y$, эксперт проверяет наличие соответствующей реализации алгоритма. Затем эксперт определяет множества функций реализации, в которых:

1) задаются параметры $\theta \in \Theta$;
2) задаются входные данные $x \in X$;
3) реализуется отображение f;
4) возвращается результат $y \in Y$.

Данные множества функций обозначаются соответственно F_1, F_2, F_3, F_4. Множества могут пересекаться или совпадать.

Для функций из множества F_1 эксперт проверяет корректность задания параметров $\theta \in \Theta$. При этом допустимым является использование в программном компоненте множества параметров Θ^*, которое является подмножеством Θ. Однако, использованное сужение множества Θ не должно состоять в ограничении области значений секретных параметров.

Для функций из множества F_2 эксперт проверяет корректность задания входных данных $x \in X$. При этом допускается, что множество входных данных X^* алгоритма является подмножеством X. Однако, использованное сужение множества входных данных должно быть оговорено в документации.

Примечание – Программа может обрабатывать не все допустимые входные данные. Например, могут шифроваться сообщения только определенной длины.

Для функций из множества F_3 эксперт проверяет тождественность отображения, реализуемого функциями, спецификации на алгоритм f (при возможных ограничениях на параметры и входные данные, использованные в реализации отображения). Для этого, по результатам анализа элементов множества F_3, составляются использованные в реализации f композиции криптографических примитивов. Затем проверяется тождественность реализованных композиций композициям криптографических примитивов, заданным в спецификации и реализующим анализируемый криптографический алгоритм. Кроме этого, эксперт проводит проверку корректности реализации вспомогательных алгоритмов, использованных в программе и не описанных в спецификации. Если такой анализ провести не удается (алгоритм не описан в документации или описан не полно, без указания использованных источников), то по данному пункту проверки выдается отрицательное заключение по причине недостаточности данных. Если использованы простые вспомогательные алгоритмы, призванные оптимизировать выполнение программы и понятные эксперту, то их описание в документации не требуется.

Для функций из множества F_4 эксперт проверяет корректность выдачи результатов $y \in Y$ выполнения криптографического алгоритма. Сужение в реализации алгоритма f множества результатов Y является недопустимым.

6.3.10 Корректность управления секретными данными

Секретные данные — это ключи, параметры и другие данные криптографических алгоритмов, значения которых в соответствии со стандартом или документацией на СКЗИ должны быть защищены от раскрытия, т.е. должны храниться в секрете.

Секретными данными СТБ 34.101.31 являются:

– ключ (включая защищаемый ключ алгоритма преобразования ключа);

– сообщение, подлежащее зашифрованию;

– результат расшифрования;

– сообщение, подлежащее имитозащите или хэшированию, если в соответствии с документацией реализация алгоритма имитозащиты или хэширования может использоваться для обработки критических данных (например, паролей в алгоритмах вычисления ключа по паролю или случайных данных в алгоритмах генерации псевдослучайных чисел).

Эксперт проверяет, что секретные данные используются в строгом соответствии с криптографическим алгоритмом. Допускается использование секретных данных во вспомогательных операциях с целью повышения быстродействия программной реализации криптоалгоритма. Другие операции с секретными данными не допускаются.

Эксперт проверяет, что все копии секретных данных в открытом виде уничтожаются при завершении работы с ними, при этом:

– значение секретных данных, размещенное в области памяти глобальной переменной, уничтожается перед каждым выходом из программы;

– значение секретных данных, размещенное в области памяти локальной переменной функции, уничтожается перед каждым выходом из данной функции;

– значение секретных данных, размещенное в динамической памяти, уничтожается перед каждым освобождением динамической памяти.

Примечание – Под уничтожением понимается такое изменение данных, хранящихся в электронных устройствах (оперативная память, память на магнитных носителях и др.), которое предотвращает их последующее восстановление. Например, уничтожение может состоять в записи в области памяти, занимаемой значениями секретных данных, фиксированных или случайно выбранных значений.

6.3.11 Отсутствие недокументированных возможностей

Эксперт определяет отсутствие недокументированных возможностей по результатам проверок, выполненных в п. 6.3.1 – 6.3.10.

Обнаруженные недокументированные возможности отражаются в протоколе анализа исходных текстов или в приложении к нему.

Приложение А
Форма протокола анализа документации

Экз. {Поле 1}

Протокол № {Поле 2} **от** {Поле 3}
результатов анализа документации
объекта испытаний {Поле 4}, реализующего криптографические алгоритмы
согласно СТБ 34.101.31-2020

1. Документы:

№	Название документа	Номер
1	{Поле 5}	{Поле 6}
2	{Поле 7}	{Поле 8}
3	{Поле 9}	{Поле 10}
4	{Поле 11}	{Поле 12}

2. При анализе документации были выполнены следующие проверки:

№	Название проверки	Отметка о выполнении
1	Проверка документа «Спецификация»	{Поле 13}
2	Проверка документа «Текст программы»	{Поле 13}
3	Проверка документа «Описание программы»	{Поле 13}
4	Проверка документа «Руководство программиста»	{Поле 13}

3. Заключение по результатам анализа документации: документация {Поле 6}, {Поле 8}, {Поле 10}, {Поле 12} соответствует (не соответствует) программе объекта испытаний в части реализации криптографических алгоритмов согласно СТБ 34.101.31-2020.

Эксперт, {Поле 15} {Поле 16}
{Поле 14}

В поле 1 указывается номер экземпляра протокола.

В поле 2 указывается номер, однозначно идентифицирующий протокол.

В поле 3 указывается дата составления протокола.

В поле 4 указывается название объекта испытаний в соответствии с представленной к испытаниям документацией.

В полях 5 и 6 указываются соответственно полное название документа «Спецификация» и его идентификационный/децимальный номер.

В полях 7 и 8 указываются соответственно полное название документа «Текст программы» и его идентификационный/децимальный номер.

В полях 9 и 10 указываются соответственно полное название документа «Описание программы» и его идентификационный/децимальный номер.

В полях 11 и 12 указываются соответственно полное название документа «Руководство программиста» и его идентификационный/децимальный номер.

В поле 13 указывается результат выполнения проверки: «положительно» — результат проверки положительный, «отрицательно» — результат проверки отрицательный. После завершения анализа документации и заполнения таблицы делается вывод о соответствии (не соответствии) документации программе объекта испытаний в части реализации криптографических алгоритмов согласно СТБ 34.101.31-2020. Вывод о соответствии делается только тогда, когда результаты всех проверок являются положительными.

В полях 14 и 16 указываются соответственно должность и Ф. И. О. эксперта.

В поле 15 ставится собственноручная подпись эксперта.

Информация об обнаруженных несоответствиях приводится в протоколе или приложении к протоколу в произвольной форме.

Приложение Б
Форма протокола тестирования

Экз. {Поле 1}

Протокол № {Поле 2} **от** {Поле 3}
результатов тестирования
объекта испытаний {Поле 4}, реализующего криптографические алгоритмы
согласно СТБ 34.101.31-2020

1. Файлы исходных текстов программ:

№	Имя файла	Хэш-значение
1	{Поле 5}	{Поле 6}
2	{Поле 5}	{Поле 6}
...	...	...

Хэш-значения для файлов вычислены согласно {Поле 7}.

2. В ходе тестирования объекта испытаний были выполнены следующие тесты:

№	Название теста	Отметка о выполнении
1	BELT.ECB.1	{Поле 8}
2	BELT.ECB.2	{Поле 8}
...	...	...

3. Заключение по результатам тестирования: объект испытаний {Поле 4} соответствует (не соответствует) требованиям, установленным в СТБ 34.101.31-2020.

Эксперт, {Поле 10} {Поле 11}
{Поле 9}

В поле 1 указывается номер экземпляра протокола.

В поле 2 указывается номер, однозначно идентифицирующий протокол.

В поле 3 указывается дата составления протокола.

В поле 4 указывается название объекта испытаний в соответствии с представленной к испытаниям документацией.

В поле 5 указываются имена исходных файлов программ объекта испытаний.

В поле 6 указывается значение функции хэширования для тестируемых файлов, вычисленное в соответствии со стандартом, указанным в поле 7. Разрешается использовать функции хэширования, определенные в СТБ 34.101.31 или СТБ 34.101.77.

В поле 8 указывается результат выполнения теста: «положительно» — тест завершен успешно, «отрицательно» — тест завершен с ошибкой; «не проводился» — тест не проводился, так как программа не поддерживает алгоритм или режим, определенный в тесте.

После завершения тестирования и заполнения таблицы делается вывод о соответствии (не соответствии) программной реализации объекта испытаний СТБ 34.101.31. Вывод о соответствии делается только тогда, когда все проводимые тесты выполнены успешно.

В полях 9, 11 указываются соответственно должность и Ф. И. О. эксперта.

В поле 10 ставится собственноручная подпись эксперта.

Приложение В
Форма протокола анализа исходных текстов

Экз. {Поле 1}

Протокол № {Поле 2} **от** {Поле 3}
результатов анализа исходных текстов программ
объекта испытаний {Поле 4}, реализующего криптографические алгоритмы
согласно СТБ 34.101.31-2020

1. Файлы исходных текстов программ:

№	Имя файла	Хэш-значение
1	{Поле 5}	{Поле 6}
2	{Поле 5}	{Поле 6}
	...	...

Хэш-значения для файлов вычислены согласно {Поле 7}.

2. В ходе анализа исходных текстов программ были выполнены следующие проверки:

№	Название проверки	Результат проверки
1	Корректность использования локальных переменных	{Поле 8}
2	Корректность использования глобальных переменных	{Поле 8}
3	Корректность использования констант	{Поле 8}
4	Корректность программной логики функций программы	{Поле 8}
5	Корректность вызова стандартных функций	{Поле 8}
6	Корректность вызова функций программы	{Поле 8}
7	Корректность обработки исключительных ситуаций	{Поле 8}
8	Корректность реализации криптографических примитивов	{Поле 8}
9	Корректность реализации криптографических алгоритмов	{Поле 8}
10	Корректность управления секретными данными	{Поле 8}
11	Отсутствие недокументированных возможностей	{Поле 8}

3. Заключение по результатам анализа исходных текстов программ: объект испытаний {Поле 4} соответствует требованиям, установленным в СТБ 34.101.31-2020.

Эксперт, {Поле 9} {Поле 10} {Поле 11}

В поле 1 указывается номер экземпляра протокола.

В поле 2 указывается номер, однозначно идентифицирующий протокол.

В поле 3 указывается дата составления протокола.

В поле 4 указывается название объекта испытаний в соответствии с представленной к испытаниям документацией.

В поле 5 указываются имена исходных файлов программ объекта испытаний.

В поле 6 указывается значение функции хэширования для исходных файлов программ, вычисленное в соответствии со стандартом, указанным в поле 7. Разрешается использовать функции хэширования, определенные в СТБ 34.101.31 или СТБ 34.101.77.

В поле 8 указывается результат выполнения проверки: «положительно» — результат проверки положительный, «отрицательно» — результат проверки отрицательный, «не проводилась» — проверка не требуется по причине специфики реализации программ объекта испытаний (например, в программе не используются глобальные переменные). После завершения анализа исходных текстов программ и заполнения таблицы делается вывод о соответствии (не соответствии) объекта испытаний СТБ 34.101.31. Вывод о соответствии делается только тогда, когда результаты всех проводимых проверок являются положительными.

В полях 9, 11 указываются соответственно должность и Ф. И. О. эксперта.

В поле 10 ставится собственноручная подпись эксперта.

Информация об обнаруженных ошибках и недокументированных возможностях приводится в протоколе или приложении к протоколу в произвольной форме и должна включать:

1) описание ошибки или недокументированной возможности;
2) имя файла и номера строк программы, содержащих ошибку.