ТОО «НИЛ «Гамма Технологии»

.

# Программное средство криптографической защиты информации «ТУМАР-CSP» версии 6

Описание программы

398-600400083267- СКЗИ 03.0.02.2-6.3.4-2020

Алматы 2020

# АННОТАЦИЯ

Настоящий документ содержит описание программы «Программного средства криптографической защиты информации «ТУМАР-CSP», версии 6.3 (далее по тексту – ПО «ТУМАР-CSP») и содержит сведения о назначении криптографического модуля, логической структуре программы, а также поддерживаемых алгоритмах и стандартах.

Документ предназначен для пользователей программы.

# ОГЛАВЛЕНИЕ

# 1 Введение

## 1.1 Наименование и обозначение

Наименование: Программное средство криптографической защиты информации «ТУМАР-CSP».

Обозначение: ПО «ТУМАР-CSP».

## 1.2 Версия ПО «ТУМАР-CSP»

ПО «ТУМАР-CSP», версии 6.3.

## 1.3 Назначение и область применения

ПО «ТУМАР-CSP» предназначено для выполнения криптографических операций в операционных системах Microsoft, управление которым происходит с помощью функций CryptoAPI.

ПО «ТУМАР-CSP» встраивается в прикладное ПО, обеспечивая хранение и обработку персональных данных, служебной, коммерческой, конфиденциальной и другой информации, не содержащей сведений, составляющих государственную тайну; обеспечивая обмен такой информацией и юридическую значимость электронного документооборота. ПО «ТУМАР-CSP» может использоваться в государственных и коммерческих структурах, а также физическими лицами.

## 1.4 Используемые сокращения

**ДСЧ** – датчик случайных чисел;
**ДКИ** – долговременная ключевая информация;
**ИОК** – Инфраструктура Открытых Ключей;
**ОП** – оперативная память;
**ОС** – операционная система;
**ПО** – программное обеспечение;
**СКЗИ** – средство криптографической защиты информации;
**УЦ** – Удостоверяющий Центр;
**ЭЦП** – электронная цифровая подпись.

# 2 Общие сведения

## 2.1 Требования к системному ПО

ПО «ТУМАР-CSP» функционирует под управлением следующих ОС

– 32/64-разрядная ОС Microsoft Windows 7/Windows 8/Windows 8.1/Windows 10;

– 64-разрядная ОС Windows Server 2008 R2/Windows Server 2012 R2/Windows Server 2016.

### 2.1.1 Требования к специализированному ПО

Для работы ПО «ТУМАР-CSP» с носителями ключевой информации необходимо установить на рабочую станцию специализированное ПО.[1]

## 2.2 Носители ключевой информации

Взаимодействие с носителями осуществляется посредством Плагинов (см. подраздел 5.6), состав которых может изменяться в зависимости от версии ПО «ТУМАР-CSP» и соответствия требованиям безопасности СТ РК 1073-2007.

### 2.2.1 Поддерживаемые носители ключевой информации

ПО «ТУМАР-CSP» обеспечивает поддержку следующих носителей ключевой информации:

– Файловая система;

– Семейство «Электронный ключ JaCarta», форм-фактор – USB/SmartCard;

– Электронный USB-ключ eToken PRO 72K (Java), форм-фактор – USB/SmartCard;

– «Электронный идентификатор «KAZTOKEN», форм-фактор – USB/SmartCard;

– SafeNet eToken 5110, форм-фактор – USB/SmartCard.

### 2.2.2 Использование носителей ключевой информации eToken PRO 72K, JaCarta, SafeNet eToken 5110

Носители ключевой информации eToken PRO 72K, JaCarta и SafeNet eToken 5110 поддерживаются ПО «ТУМАР-CSP» только при использовании Плагинов, разработанных ТОО «НИЛ «Гамма Технологии».

## 2.3 Используемые генераторы случайных чисел

ПО «ТУМАР-CSP» обеспечивает поддержку следующих генераторов случайных чисел:

[1] ПО для работы с носителями ключевой информации устанавливается при установке ПО «ТУМАР-CSP». См. руководство по установке

– аппаратный генератор eToken PRO 72K, форм-фактор – USB/SmartCard;
– аппаратный генератор JaCarta, форм-фактор – USB/SmartCard;
– аппаратный генератор SafeNet eToken 5110, форм-фактор – USB/SmartCard.

Данные источники случайности используются на различных этапах работы ПО «ТУМАР-CSP»:

– генерация личных (секретных) составляющих криптографических ключей;
– генерация секретного параметра при вычислении ЭЦП;
– выдача случайности по запросу пользовательского приложения;
– генерация ключа шифрования пользовательских данных;
– генерация сеансовых ключей шифрования, обеспечивающих конфиденциальность передаваемых данных между приложением и устройством.

# 3 Условия применения

## 3.1 Алгоритмы, соответствующие третьему уровню безопасности

Алгоритм ГОСТ генерации криптографических ключей, шифрования, выработки и проверки ЭЦП, в соответствии с ГОСТ 34.310-2004.

Длина открытого ключа – 512 бит, длина закрытого ключа 256 бит.

Алгоритм RSA генерации криптографических ключей, шифрования, выработки и проверки ЭЦП.

Длина открытого ключа – 4096 бит, длина закрытого ключа – 4096 бит.

При генерации и формирования ключей обеспечивается принятие каждым битом ключа единичного значения с вероятностью из интервала (0,50 ± 0,003).

Хеширование данных по алгоритмам SHA-2 и ГОСТ 34.311-95:

– 256 бит (ГОСТ);

– SHA-256;

– SHA-384;

– SHA-512.

Шифрование данных в соответствии с ГОСТ 28147-89 во всех режимах, предусмотренных стандартом.

Длина ключа 256 бит.

Имитационная защита данных в соответствии с ГОСТ 28147-89.

Длина имитовставки 64 бит.

## 3.2 Поддерживаемые стандарты

1. ГОСТ 28147-89 «Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования».
2. RSA Cryptography Standard (Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications).
3. Криптографические алгоритмы SHA-2:
    3.1. Federal Information Processing Standards Publication 180-2 Secure Hash Standard (SHS), August 2002;
    3.2. Federal Information Processing Standards Publication 180-2 Secure Hash Standard (SHS) with a change notice, February 2004;
    3.3. Federal Information Processing Standards Publication 180-3 Secure Hash Standard (SHS) with a change notice, October 2008.
4. ГОСТ 34.310-2004 «Информационная технология. Криптографическая защита

информации. Процессы формирования и проверки электронной цифровой подписи».

5. ГОСТ 34.311-95 «Информационная технология. Криптографическая защита информации. Функция хэширования».

6. СТ РК 1073-2007 «Средства криптографической защиты информации. Общие технические требования».

### 3.3 Ограничение использования

В силу особенностей архитектуры ОС Windows, если на компьютере уже установлены криптопровайдеры других производителей, реализующие криптографические алгоритмы ГОСТ, корректная работа ПО «ТУМАР-CSP» не гарантируется.

Для корректной работы ПО «ТУМАР-CSP» под различными учетными записями необходимо производить его установку, войдя в систему под именем пользователя. В случае если пользователь не обладает правами администратора системы, то необходимо временно повысить его полномочия.

### 3.4 Совместная работа с антивирусными программами

#### 3.4.1 Работа с ПО «Антивирус Касперского»

Для корректной работы ПО «ТУМАР-CSP» с ПО «Антивирус Касперского» рекомендуется установить программы в следующей последовательности:

1. Установить пакет дистрибутивов ПО «Антивирус Касперского»/осуществить настройку режима «Превентивная защита» в ПО «Dr.Web Security Space».
2. Установить ПО «ТУМАР-CSP».

**Внимание**: Перед установкой ПО «ТУМАР-CSP» рекомендуется отключить антивирусную программу. Если по каким-либо причинам данное действие невозможно, то существует вероятность блокировки инсталляционных файлов. В случае блокирования инсталлятора необходимо добавить дистрибутив **SetupCSPv6.3.exe** в доверенную зону антивирусного приложения и повторить инсталляцию ПО «ТУМАР-CSP».

#### 3.4.2 Работа с ПО «Dr.Web Security Space»

Для корректной установки ПО «ТУМАР-CSP» на рабочую станцию, на которой установлено ПО «Dr.Web Security Space» актуальной версии, необходимо в настройках режима «Превентивная защита» ПО «Dr.Web Security Space» разрешить автозапуск оболочки Windows.

Для этого:

1. Вызвать контекстное меню, нажав правой клавишей мыши по иконке ПО, расположенной в systray. Выбрать пункт меню **Инструменты → Настройки**.

2. В окне **Настройки → Основные** перейти на закладку **Превентивная защита** и в рабочей области окна нажать на кнопку **Пользовательский**.

3. В окне **Превентивная защита** для опции **Автозапуск оболочки Windows** выбрать

значение **Разрешать**.

4. Принять внесенные изменения в настройки ПО «Dr.Web Security Space».

5. Осуществить установку ПО «ТУМАР-CSP» на рабочую станцию.

### 3.4.3 Работа с ПО «McAfee»

При работе с антивирусной программой McAfee, в ее настройках необходимо отключить опцию **Buffer overflow protection**.

## 3.5 Работа с ПО «SWIFT Alliance»

Для корректного взаимодействия ПО «ТУМАР-CSP» и ПО «SWIFT Alliance» необходимо, чтобы установка ПО «ТУМАР-CSP» производилась после установки ПО «SWIFT Alliance» и при условии активности основного процесса ПО «SWIFT Alliance». В случае если ПО «ТУМАР-CSP» уже установлено на компьютере, требуется:

- удалить ПО «ТУМАР-CSP»;
- установить, настроить и запустить ПО «SWIFT Alliance»;
- установить ПО «ТУМАР-CSP».

# 4 Функциональное назначение

ПО «ТУМАР-CSP» предназначено для криптографической обработки информации, генерации ключевой информации для симметричного и ассиметричного шифрования, предоставляя возможность решать следующие задачи:

– авторизация и обеспечение юридической значимости электронных документов посредством использования процедур формирования и проверки ЭЦП;

– обеспечение конфиденциальности и контроля целостности информации посредством ее шифрования и имитозащиты;

– создание криптографически стойких ключевых пар и сессионных ключей;

– обеспечение аутентификации связывающихся сторон.

ПО «ТУМАР-CSP» состоит из Конфигуратора, Исполнимого модуля и Библиотеки разработчика, построенной в соответствии с криптографическим интерфейсом компании Microsoft – Cryptographic Service Provider (CSP).

Разработчики имеют возможность подключать ПО «ТУМАР-CSP» в приложения верхнего уровня для криптографической обработки пользовательских данных и, таким образом, получать доступ к криптографическим функциям по стандартному программному интерфейсу. При этом ПО «ТУМАР-CSP» обеспечивает изолированность ключевой системы и криптографических алгоритмов от пользовательских данных, что исключает возможность компрометации ключевой системы и/или изменения параметров криптографических алгоритмов. Во время работы ПО «ТУМАР-CSP» предоставлена возможность анализа результатов работы каждой функции для диагностики возникающих ошибок. Результат работы функций возвращается через файловую систему и ОП в виде значения хэш-функции, зашифрованных данных, ЭЦП или результатов проверки ЭЦП. Дополнительно ПО «ТУМАР-CSP» позволяет указывать длину секретных и открытых ключей, а также их взаимосвязь для алгоритмов асимметричного шифрования.

ПО «ТУМАР-CSP» ориентировано на использование механизма открытого распределения ключей на базе асимметричного алгоритма, предложенного в 1976 году учеными У. Диффи и М. Хеллманом.

В качестве средства управления ключевой информацией участников криптообмена рекомендуется использование механизмов, реализующих ИОК. Само ПО «ТУМАР-CSP» механизмы ИОК не реализует.

## 4.1 Основные функции

Для защиты информации ПО «ТУМАР-CSP» предоставляет следующие возможности:

– генерация ключевых пар в соответствии с ГОСТ 34.310-2004 для формирования подписи и ключевого обмена;

– генерация ключевых пар в соответствии с криптографическим алгоритмом RSA для формирования ЭЦП, шифрования и аутентификации;

– формирование и проверка ЭЦП;

– шифрование данных в соответствии с ГОСТ 28147-89 во всех режимах, предусмотренных стандартом;

– хэширование данных в соответствии с ГОСТ 34.311-95 и SHA (FIPS PUB 180-2/3);

– имитационная защита данных в соответствии с ГОСТ 28147-89;

– генерация закрытых ключей на носители ключевой информации;

– использование пароля для доступа к контейнеру с ключевой информацией для создания дополнительных мер защиты;

– хранение сертификатов открытых ключей пользователей в ключевом контейнере;

– хранение цепочек сертификатов в ключевом контейнере;

– работа с ключевым контейнером в режиме копирования ключевой информации, импорта и экспорта сертификата открытого ключа пользователя и УЦ; проверки/смены статуса сертификата, депонирования ключевой информации пользователя на сервере УЦ, установки сертификата открытого ключа пользователя в хранилище ОС Windows с привязкой к секретному ключу.

# 5 Описание логической структуры

## 5.1 Состав ПО «ТУМАР-CSP»

В состав ПО «ТУМАР-CSP» входят следующие компоненты:

– исполнимый модуль **cptumar.exe**, выполняющий операции хэширования, имитозащиты, шифрования, формирования и проверки ЭЦП (далее по тексту Исполнимый модуль)

– библиотека **cptumar.dll**, содержащая набор функций, соответствующих криптографическому интерфейсу фирмы Microsoft – Cryptographic Service Provider (CSP), (далее по тексту – Криптопровайдер);

– библиотека **TUMEXT.dll**, содержащая набор функций по взаимодействию со средой Windows через интерфейс CryptoAPI для поддержки работы с ассиметричными/симметричными ключами и сертификатами;

– конфигуратор ключевой информации **tumarconf.exe**, позволяющий генерировать, вводить в действие, копировать, удалять ДКИ и управлять настройками доступа к ключам, а также выбирать алгоритмы и длины ключей (далее по тексту – Конфигуратор);

– Плагины (дополнительные библиотеки), обеспечивающие защищенный доступ к ключам, размещенным на различных носителях ключевой информации, и использование устройств в качестве ДСЧ.

Все компоненты написаны на языке C++ с использованием компиляторов Visual C++ 6.0 и Borland C++ 5.5.1 for Win32.

## 5.2 Исполнимый модуль

Исполнимый модуль – **cptumar.exe** скомпилирован в виде консольного приложения, предоставляющего удобный интерфейс для вызова криптографических функций. Исполнимый модуль не требует дополнительной настройки и не хранит параметры, необходимые для работы на файловой системе или в реестре ОС Windows.

Входными/выходными данными являются имена файлов, содержащих открытые и зашифрованные сессионные ключи, значения хэш-функций, имитовставки, ЭЦП и пользовательские данные. Дополнительно, в качестве входных данных, могут использоваться номера алгоритмов шифрования, вычисления хэш-функции и имитовставки. Если алгоритмы не указаны, то используются значения по умолчанию: для шифрования – ГОСТ 28147-89, для хэширования – ГОСТ 34.311–95, для ЭЦП – ГОСТ 34.310-2004, в зависимости от выбранных параметров ключевой информации.

Во время работы Исполнимый модуль получает пользовательские данные от файловой системы и передает их на обработку криптографическим функциям из библиотеки Криптопровайдера (см. раздел 6.3), в соответствии с криптографическим интерфейсом Cryptographic Service Provider. Полученные результирующие данные записываются в виде файлов, результат выполнения самой функции выводится на дисплей в виде кода возврата.

### 5.3 Криптопровайдер

Криптопровайдер скомпилирован в виде библиотеки (cptumar.dll), которая может быть использована в приложениях верхнего уровня. При загрузке библиотека производит самопроверку собственного кода и кода загружаемых Плагинов для выявления несанкционированной модификации. В случае отрицательного результата проверки собственного кода, возвращается сообщение об ошибке и дальнейшая работа с библиотекой невозможна. В случае отрицательного результата проверки кода Плагинов, библиотеки игнорируются.

Во время работы Криптопровайдер загружает ДКИ, используя Плагины и пользовательские настройки, установленные с помощью Конфигуратора. Сама ДКИ хранится в виде ключевого контейнера. При этом целостность контейнера обеспечивается использованием значения хэш-функции для ключей, длина которых менее 336 бит криптографического преобразования на эллиптических кривых, и имитовставки для ключей, длина которых более 336 бит, прописанным в самом контейнере, а секретность – паролем, который вводится в диалоговом окне. Во время загрузки происходит проверка на целостность.

Для зашифровывания данных Криптопровайдер создает сессионные ключи: ДСЧ генерирует последовательность псевдослучайных чисел. Сессионные ключи извлекаются из Криптопровайдера только в зашифрованном виде. Открытые ключи извлекаются из Криптопровайдера в незащищенном виде. В качестве способа безопасного распределения открытых ключей рекомендуется использование ИОК. Настройки ДКИ (далее – Профайлы) хранятся в специальном конфигурационном файле cptumar.conf.

Входными данными для Криптопровайдера являются: пользовательские данные, настройки из Профайлов, получаемые посредством Плагинов ДКИ. Выходными данными для Криптопровайдера являются: пользовательские данные, возвращаемые через ОП, и данные, передаваемые Плагинам для обработки.

### 5.4 Конфигуратор

Для панели управления ОС Windows Конфигуратор (**tumarconf.exe**) скомпилирован в виде апплета с графическим пользовательским интерфейсом. Конфигуратор построен в виде диалогового окна, состоящего из панели настроек – списка Профайлов, и контекстно-зависимой информационной части, в которой отображаются свойства ДКИ.

Конфигуратор позволяет устанавливать параметры ДКИ, включая алгоритмы, пароль доступа к ДКИ, выполнять операции генерации, копирования, удаления ДКИ. Поддерживается следующий метод вычисления открытого ключа по сформированному секретному: алгоритм Диффи-Хелмана на эллиптических кривых. Во время работы Конфигуратор получает доступ к ДКИ через Плагины. Параметры, необходимые Конфигуратору для работы, а также настройки, выставленные во время его работы, хранятся в специальном конфигурационном файле *cptumar.conf*.

Входными данными для Конфигуратора являются: информация, введенная с клавиатуры и настройки из файла *cptumar.conf*, получаемые посредством Плагинов ДКИ.

Выходными данными для Конфигуратора являются: выводимая на экран графическая и

текстовая информация, записи в файл cptumar.conf, данные, передаваемые Плагинам для обработки.

### 5.5 Библиотека TUMEXT.dll

Модуль взаимодействия со средой ОС Windows через интерфейс CryptoAPI скомпилирован в виде DLL–библиотеки – **TUMEXT.dll**, предназначен для поддержки работы с ассиметричными и симметричными ключами.

Выходными данными для библиотеки **TUMEXT.dll** является экспорт списка OID-ов (Original Issue Discount) из реестра ОС Windows.

### 5.6 Плагины

Плагины скомпилированы в виде библиотек, предоставляющих интерфейс для чтения/записи/удаления информации с различных устройств долговременного хранения: жесткие диски, USB-ключи и т.д, и используются Криптопровайдером и Конфигуратором для управления ДКИ. Код Плагина при загрузке Криптопровайдера проверяется для выявления несанкционированной модификации и, в случае отрицательного результата проверки, игнорируется. При наличии соответствующих настроек, используется шифрование/ расшифрование ключевого контейнера.

Входными данными для Плагинов являются: ДКИ, получаемая от Криптопровайдера и/или Конфигуратора. Выходными данными для Плагинов являются: ДКИ, передаваемая Криптопровайдеру и/или Конфигуратору для дальнейшего использования.

**Внимание:** При использовании Плагинов сторонних разработчиков (например, устройств генерации и хранения ключевой информации), которые не входят в состав дистрибутива ПО «ТУМАР-CSP», корректная работа с данными устройствами в информационных системах, использующих «ТУМАР-CSP», не гарантируется. Разработчик ПО «ТУМАР-CSP» не несет ответственности за появление сбоев в работе информационных систем и иных проблем, которые могут быть вызваны использованием сторонних устройств и их объектных модулей (*.dll).

# 6 Описание использования

## 6.1 Запуск ПО

Запуск Исполнимого модуля осуществляется набором в командной строке имени **cptumar.exe** с соответствующими параметрами.

Запуск Конфигуратора производится путем вызова ПО «TumarCSP Конфигуратор» через Панель Управления ОС Windows.

Для загрузки Криптопровайдера **cptumar.dll** используется стандартная команда `LoadLibrary`.

Самостоятельная загрузка Плагинов не требуется.

## 6.2 Загрузка ключей

Две пары ключей: секретный и открытый ключ для подписи, секретный и открытый ключ для шифрования составляют ДКИ участника криптообмена. Для выполнения операций, связанных с шифрованием информации и формированием подписи, необходима загрузка ключевой системы. Загрузка ключевой системы участника криптообмена осуществляется при помощи функции **CPAcquireContext** из библиотеки Криптопровайдера. При этом осуществляется проверка целостности ДКИ с использованием значения хэш-функции на ключевой контейнер. Перед завершением работы модуля защиты или при перезагрузке ключевой системы, необходимо выполнить процедуру **CPReleaseContext**, которая удаляет остаточную ключевую информацию и корректно освобождает память.

Во время работы исполнимого модуля **cptumar.exe** загрузка ДКИ производится в следующих командах: вычисления и проверки имитовставки (**-mac** и **-chkm**), зашифрования/расшифрования (**-crp** и **-uncr**) и формирования ЭЦП (**-sgn**).

## 6.3 Генерация ключей

Генерация ДКИ производится с помощью вызова **CPAcquireContext** в режиме создания ключей. При этом секретные ключи генерируются с помощью аппаратного ДСЧ (см. раздел 2.3). Полученные последовательности проверяются на равновероятное распределение 0 и 1 по методу хи-квадрат. При этом вероятность единичного значения лежит в интервале (в соответствии с СТ РК 1073-2007): (0,500 ± 0,003) – для 3-го уровня безопасности;

Далее, статистически «хорошие» последовательности перемешиваются с использованием одноканальной линии задержки, и могут быть использованы в качестве секретных ключей. Соответствующие им открытые ключи вычисляются в соответствии со схемой Диффи-Хелмана над конечным алгебраическим полем или в соответствии с математическим аппаратом эллиптических кривых. Размер ключей, а также параметры и методы для вычисления открытого ключа по сформированному секретному ключу, задаются Конфигуратором.

Полученная ДКИ сохраняется на ключевом носителе в виде ключевого контейнера, включающего в себя ключевые пары, время создания, алгоритм формирования открытого

ключа и хэш-значение на ключевой контейнер.

Генерация ключа сессии, использующегося при шифровании, производится с помощью вызова функции **CPGenKey** из библиотеки Криптопровайдера. При этом генерируется только один ключ, который будет использован для симметричного шифрования. Во время использования исполнимого модуля **cptumar.exe** генерация ключей осуществляется по команде **-gen**.

### 6.4 Импорт/экспорт ключевой информации

При проверке ЭЦП используется открытый ключ автора сообщения. Это порождает задачу экспортирования/импортирования открытого ключа из/в Криптопровайдер для обеспечения функции проверки ЭЦП. Сам открытый ключ извлекается в незащищенном виде в формате PUBLICKEYBLOB. Для обеспечения подлинности и целостности открытого ключа при передаче по каналам связи рекомендуется оформлять открытый ключ в виде электронного сертификата, подписанного на ключах доверенного центра. Такая возможность обеспечивается системами, построенными на базе ИОК.

Для выполнения функций шифрования и/или имитозащиты используется симметричный секретный ключ, который необходимо безопасным способом передать получателю сообщения. Требуется выполнять операции экспортирования/импортирования сессионного ключа из/в Криптопровайдер в зашифрованном виде в формате SIMPLEBLOB. В качестве алгоритма шифрования применяется ассиметричный алгоритм, использующий секретный ключ того, кто шифровал, и открытый ключ того, для кого шифровали. Это в свою очередь порождает необходимость экспортирования/импортирования открытого ключа из/в Криптопровайдер.

Эти обе задачи решаются с использованием функций **CPExportKey** и **CPImportKey** из библиотеки Криптопровайдера

Во время использования исполнимого модуля **cptumar.exe** импорт/экспорт открытых ключей осуществляется в следующих командах – вычисления и проверки имитовставки (**-mac** и **-chkm**), зашифрования/расшифрования (**-crp** и **-uncr**) и проверки ЭЦП (**-vrfs**).

### 6.5 Распределение ключевой информации

Для распределения ключей симметричного шифрования и/или имитозащиты используется возможность шифрования на ассиметричном алгоритме, после которой их можно распространять по открытым каналам связи. ПО «ТУМАР-CSP» не позволяет осуществлять функции экспорта/импорта таких ключей в незащищенном виде.

Для распределения открытых ключей ассиметричного шифрования (в том числе ключей для ЭЦП), рекомендуется использование ИОК. ПО «ТУМАР-CSP» не содержит в себе функциональных возможностей ИОК.

### 6.6 Операция шифрования/расшифрования

Для обеспечения конфиденциальности электронной информации, хранимой или

передаваемой по каналам связи, используется шифрование. В ПО «ТУМАР-CSP» реализован алгоритм шифрования в соответствии с ГОСТ 28147-89. Алгоритм работает в режиме гаммирования с обратной связью на симметричном ключе.

Для зашифрования информации используется функция **CPEncrypt**. Для расшифровывания информации используется функция **CPDecrypt**. Функции **CPExportKey** и **CPImportKey** обеспечивают возможность, соответственно:

– экспорта из Криптопровайдера симметричного ключа шифрования, защищенного на секретном ключе отправителя и открытом ключе получателя сообщения;

– импорта в Криптопровайдер симметричного ключа шифрования, который расшифруется с использованием открытого ключа отправителя и секретного ключа получателя сообщения.

Во время использования исполнимого модуля **cptumar.exe,** зашифрование/ расшифрование пользовательских данных осуществляется, соответственно, в следующих командах: **-crp** и **-uncr**.

### 6.7 Операция хэширования

Механизм хэширования реализован в соответствии с ГОСТ 34.311-95.

Для вычисления значения хэш-функции используются следующие функции из библиотеки Криптопровайдера:

**CPCreateHash** – создание объекта;

**CPHashData** – хэширование данных;

**CPGetHashParam** – получение значения хэш-функции.

Во время использования исполнимого модуля **cptumar.exe** вычисление/проверка значения хэш-функции на пользовательские данные осуществляется, соответственно, в следующих командах: **-hsh** и **-chkh**.

### 6.8 Операция вычисления и проверки имитовставки

Степень защищенности информации от случайного или умышленного внесения изменений (искажений) называется имитостойкостью. Для обеспечения имитостойкости электронной информации, хранимой или передаваемой по каналам связи, используется имитозащита.

В ПО «ТУМАР-CSP» имитозащита реализована в соответствии с алгоритмом ГОСТ 28147-89. Алгоритм работает в режиме вычисления имитовставки на симметричном ключе.

Для вычисления и проверки имитовставки используются следующие функции из библиотеки Криптопровайдера:

**CPCreateHash** – создание объекта;

**CPHashData** – хэширование данных;

**CPImportKey** – экспорт из Криптопровайдера симметричного ключа имитозащиты, защищенного на секретном ключе отправителя и открытом ключе получателя сообщения;

**CPExportKey** – импорт в Криптопровайдер симметричного ключа имитозащиты, который расшифровывается с использованием открытого ключа отправителя и секретного

ключа получателя сообщения;

**CPGetHashParam** – получение значения хэш-функции.

Во время использования исполнимого модуля **cptumar.exe,** вычисление/проверка имитовставки на пользовательские данные осуществляется, соответственно, в следующих командах: **-mac** и **-chkm**.

### 6.9 Операция формирования/проверки ЭЦП

Для обеспечения целостности, достоверности и юридической значимости электронных документов используется механизм ЭЦП. Поставить ЭЦП может только автор документа, имеющий секретный ключ. Осуществить проверку может любой участник системы защиты, имеющий открытый ключ автора документа.

В ПО «ТУМАР-CSP» механизм ЭЦП реализован в соответствии с алгоритмом ГОСТ 34.310–2004.

Для вычисления ЭЦП электронного документа, вначале вычисляется хэш-функция от открытого текста и используется функция **CPSignHash** из библиотеки Криптопровайдера. Для проверки ЭЦП электронного документа используется функция **CPVerifySignature**.

При формировании ЭЦП для ключей длиной не менее 512 бит криптографического преобразования на эллиптических кривых используется алгоритм вычисления хэш-функции SHA2. Для прочих длин ключей рекомендуется использовать значение по умолчанию.

Во время использования исполнимого модуля **cptumar.exe,** формирование/проверка ЭЦП на пользовательские данные осуществляется, соответственно, в следующих командах: **-sgn** и **-vrfs**.