127 018, Москва, Сущевский Вал, 18
Телефон: (495) 995 4820
Факс: (495) 995 4820
http://www.CryptoPro.ru
E-mail: info@CryptoPro.ru

Средство Криптографической Защиты Информации

КриптоПро JCP

Версия 2.0 R2

Описание реализации

ЖТЯИ.00091-02 91 01-01

Листов 27

2018

Оглавление

1. Аннотация

Настоящий документ содержит описание реализации средства криптографической защиты информации «КриптоПро JCP» версия 2.0 R2 (далее - СКЗИ) и сведения о текущем состоянии продукта.

2. Назначение СКЗИ

СКЗИ предназначено для защиты открытой информации в информационных системах общего пользования (вычисление/проверка электронной подписи) и защиты конфиденциальной информации, не содержащей сведений, составляющих государственную тайну, в корпоративных информационных системах с выполнением следующих функций:

- авторизацию электронных документов на базе электронной подписи;
- аутентификацию сторон при передаче электронных документов на базе протоколов TLS;
- защищенную парно-выборочную связь для обмена конфиденциальной информацией.

СКЗИ обеспечивает выполнение следующих функций:

- авторизация и обеспечение юридической значимости электронных документов при обмене ими между пользователями посредством использования процедур формирования и проверки (с использованием сертификатов стандарта X.509 Удостоверяющего центра) электронной подписи в соответствии с отечественными стандартами (RFC 4357):

ГОСТ Р 34.10-2001. *"Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи".*

ГОСТ Р 34.10-2012. *"Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи".*

ГОСТ Р 34.11-94. *"Информационная технология. Криптографическая защита информации. Функция хэширования"*.

ГОСТ Р 34.11-2012. *"Информационная технология. Криптографическая защита информации. Функция хэширования"*.

- обеспечение конфиденциальности и контроля целостности информации посредством ее шифрования и имитозащиты, в соответствии с отечественным стандартом:
- ГОСТ 2814789 *"Системы обработки информации. Защита криптографическая"*;
- контроль целостности системного и прикладного программного обеспечения для его защиты от несанкционированного изменения или от нарушения правильности функционирования;
- управления ключевыми элементами системы в соответствии с регламентом;

обеспечение аутентификации связывающихся сторон, конфиденциальности и целостности пересылаемой информации с использованием сертификатов стандарта X.509;

- установление аутентичного защищенного соединения с использованием протокола КриптоПро JTLS;
- обеспечение конфиденциальности и контроля целостности и авторизация файлов и информационных сообщений;
- обеспечение аутентификации, аутентификация пользователя в домене Windows.
- Дополнительные алгоритмы поддержки ключевых систем, параметры алгоритмов, форматы сертификатов, поддерживаемые в СКЗИ, определены в документах RFC 4357, RFC 4490, RFC 4491.

Дополнительные алгоритмы поддержки ключевых систем, параметры алгоритмов, форматы сертификатов, поддерживаемые в СКЗИ, определены в документах RFC 4357, RFC 4490, RFC 4491, «Системы обработки информации. Защита криптографическая. Методические рекомендации по криптографическим алгоритмам, сопутствующим применению стандартов ГОСТ Р 34.10-2012 и ГОСТ Р 34.11-2012» Технический комитет по стандартизации «Криптографическая защита информации» (ТК 26), «Информационная технология. Криптографическая защита информации. Рекомендации по стандартизации. Использование алгоритмов ГОСТ 28147-89, ГОСТ Р 34.11 и ГОСТ Р 34.10 в криптографических сообщениях формата CMS» Технический комитет по стандартизации «Криптографическая защита информации» (ТК 26), «Техническая спецификация. Использование алгоритмов ГОСТ Р 34.10, ГОСТ Р 34.11 в профиле сертификата и списке отзыва сертификатов (CRL) инфраструктуры открытых ключей X.509» Технический комитет по стандартизации «Криптографическая защита информации» (ТК 26).

Допустимо использовать следующие механизмы защиты информации:

- Конфиденциальность информации при хранении (на дисках, в базе данных) и передаче в сети связи обеспечивается использованием функций шифрования.

- Идентификация и авторство. При сетевом взаимодействии (установлении сеанса связи) обеспечивается функциями ЭП при использовании их в процессе аутентификации (например, в соответствии с рекомендациями X.509). При электронном документообороте обеспечивается использованием функций ЭП электронного документа. Дополнительно должна быть предусмотрена защита от навязывания, повтора электронного документа и целостность справочников ключей проверки ЭП.

- Целостность информации. Обеспечивается использованием функций ЭП электронного документа. При использовании функций шифрования (без использования ЭП) обеспечивается имитозащитой. Для обеспечения целостности хранимых данных может быть использована функция хэширования или имитозащиты, но при этом не обеспечивается авторство информации.

- Неотказуемость от передачи электронного документа. Обеспечивается использованием функций ЭП (подпись документа отправителем) и хранением документа с ЭП в течение установленного срока приемной стороной.

- Неотказуемость от приема электронного документа. Обеспечивается использованием функций ЭП и квитированием приема документа (подпись квитанции получателем), хранением документа и квитанции с ЭП в течение установленного срока отправляющей стороной.

- Защита от переповторов. Обеспечивается использованием криптографических функций ЭП, шифрования или имитозащиты с добавлением уникального идентификатора сетевой сессии (электронного документа) с последующей их проверкой приемной стороной или разработкой специализированного протокола аутентификации (обмена электронными документами).

- Защита от навязывания информации. Защита от нарушителя с целью навязывания им приемной стороне собственной информации, переданной якобы от лица санкционированного пользователя (нарушение авторства информации). Обеспечивается использованием функций ЭП с проверкой атрибутов электронного документа и ключа проверки ЭП отправителя.

- Защита от закладок, вирусов, модификации системного и прикладного ПО обеспечивается совместным использованием криптографических средств, средств антивирусной защиты и организационных мероприятий.

3. Программно-аппаратные среды функционирования СКЗИ

СКЗИ предназначено для использования на следующих программно-аппаратных платформах:

Windows

Windows Vista/7/8/8.1/10 (x86, x64) (только совместно с Java-машиной производства Oracle);

Windows Server 2003/2008/2008 R2/2012/2012 R2/2016 (x64) (только совместно с Java-машиной производства Oracle);

LSB Linux

ОС Linux, удовлетворяющие стандарту Linux Standart Base ISO/IEC 23360 (x86, x64) версии LSB 4.x:

CentOS 4/5/6/7 (x86, x64);

ТД ОС АИС ФССП России (GosLinux) (x86, x64);

Red OS (x86, x64);

Fedora 23/24/25/26/27 (x86, x64);

Mandriva Enterprise Server 5, Business Server 1 (x86, x64);

Oracle Linux 5/6/7 (x86, x64);

OpenSUSE 12.2/12.3/13.1/13.2 (x86, x64);

SUSE Linux Enterprise 10/11/12 (x86, x64, POWER);

Red Hat Enterprise Linux 4/5/6/7 (x86, x64, POWER);

Ubuntu 10.04/12.04/12.10/13.04/14.04/14.10/15.04/15.10/16.04/16.04.1/16.10/17.04/17.10/18.04 (x86, x64, POWER);

Linux Mint 13/14/15/16/17/18 (x86, x64);

Debian 7/8/9 (x86, x64, POWER);

Unix

ALT Linux 6/7 (x86, x64);

ALT Linux 6/7 (ARM) (только совместно с Java-машиной производства Oracle);

Ubuntu Phone (ARM) (только совместно с Java-машиной производства Oracle);

ROSA 2011, Enterprise Desktop X.1 (Marathon), Enterprise Linux Server (x86, x64);

РОСА ХРОМ/КОБАЛЬТ/НИКЕЛЬ (x86, x64);

FreeBSD 8/9/10/11/pfSense 2.x (x86, x64);

AIX 5/6/7 (POWER) (только совместно с Java-машиной производства IBM);

Mac OS X 10.7/10.8/10.9/10.10/10.11/10.12 (x64);

Solaris 10 (sparc, x86, x64) (только совместно с Java-машиной производства Oracle);

Solaris 11 (sparc, x64) (только совместно с Java-машиной производства Oracle).

4. Основные характеристики СКЗИ

4.1. Размеры ключей

Размеры ключей электронной подписи:

- ключ электронной подписи 256 бит или 512 бит;
- ключ проверки электронной подписи 512 бит или 1024 бита.

Размеры ключей, используемых при шифровании:

- закрытый ключ 256 бит или 512 бит;
- открытый ключ 512 бит или 1024 бита;
- симметричный ключ 256 бит.

4.2. Типы ключевых носителей

Используются ключевые носители:

1. ГМД 3,5’’, USB диски;
2. eToken, Jacarta;
3. Rutoken;
4. Смарткарты Оскар;
5. ESMART Token;
6. Раздел HDD ПЭВМ.

Использование ключевых носителей в зависимости от программно-аппаратной платформы отражено в ЖТЯИ.00091-02 90 01-01. «Руководство администратора безопасности».

1. В состав дистрибутива СКЗИ входят библиотеки для интеграции ключевых носителей в «КриптоПро JCP» версия 2.0 R2, но не входят модули поддержки и драйвера для ОС. По вопросам получения модулей поддержки и драйверов необходимо обращаться к производителям соответствующих устройств.
2. Хранение закрытых ключей на HDD ПЭВМ (в разделе HDD) допускается только при условии распространения на HDD или на ПЭВМ с HDD требований по обращению с ключевыми носителями (ЖТЯИ.00091-01 90 01-01. Руководство администратора безопасности общая часть).
3. Все вышеперечисленные носители используются только в качестве пассивного

	хранилища ключевой информации без использования криптографических механизмов, реализованных на смарт-карте/токене. 4. Использование носителей других типов допускается только по согласованию с ФСБ России.

5. Структура и состав СКЗИ

5.1. Структура СКЗИ

Общая структура СКЗИ представлена на Рисунке 1.

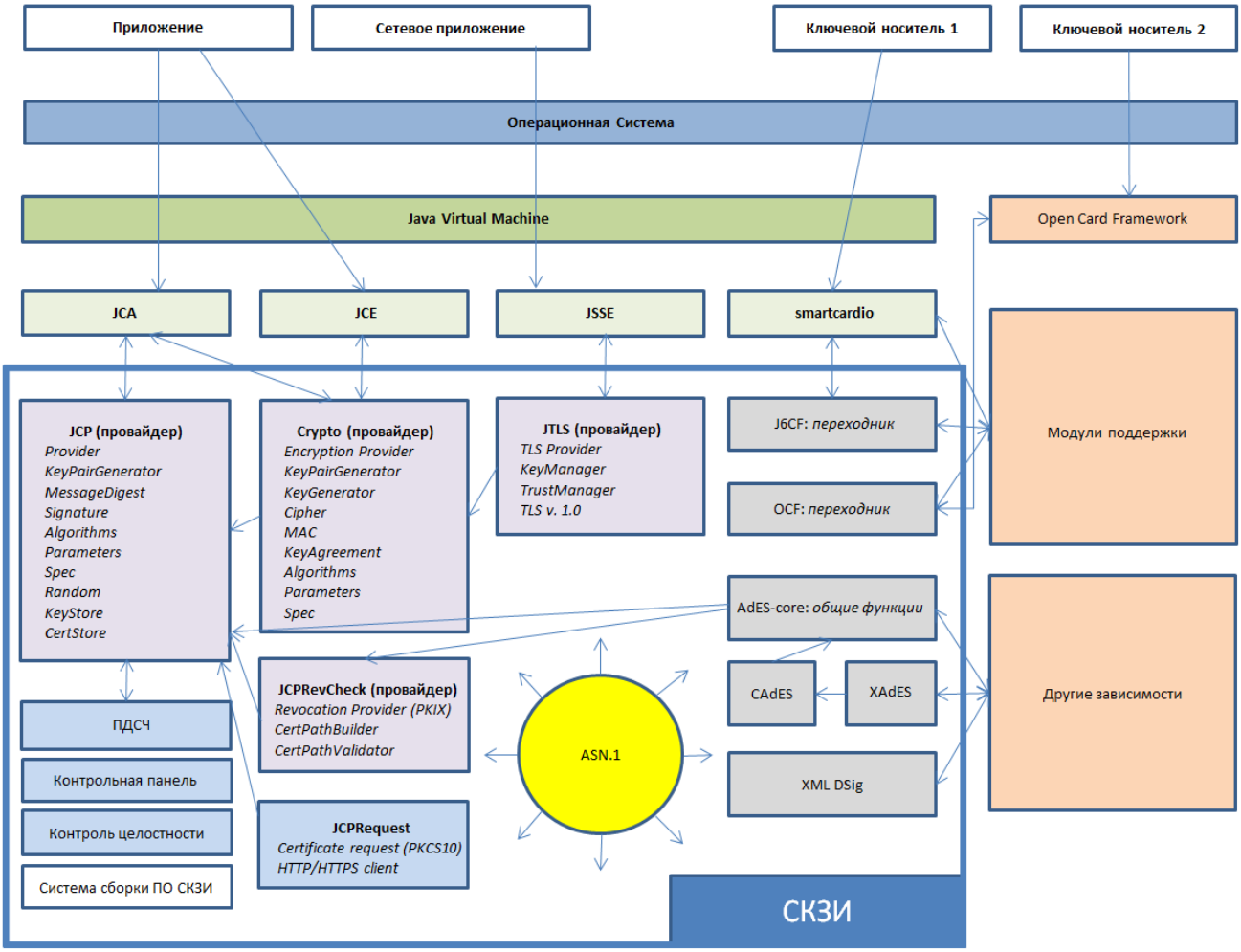

Рисунок 1 – Структура СКЗИ «КриптоПро JCP» версия 2.0 R2.

5.2. Исполнения СКЗИ

СКЗИ ЖТЯИ.00091-02 выпускается по классу защиты КС1 для Java-машины.

Исполнение 1 класса защиты КС1 выполнено в составе:

криптопровайдер (модуль на прикладном уровне);

модуль обработки сертификатов и CMS протокола;

библиотека, обеспечивающая подключение и функционирование ключевых носителей (RDK);

модуль поддержки интерфейса под управлением виртуальной Java-машины,

набор модулей и Java-классов для поддержки Java JCA интерфейса.

и функционирует в группах программно-аппаратных сред в соответствии с п.2.

Исполнение 2 класса защиты КС1 выполнено в составе:

криптопровайдер (модуль на прикладном уровне);

модуль шифрования;

модуль сетевой аутентификации (КриптоПро JTLS);

модуль обработки сертификатов и CMS протокола;

библиотека, обеспечивающая подключение и функционирование ключевых носителей (RDK);

модуль поддержки интерфейса под управлением виртуальной Java-машины,

набор модулей и Java-классов для поддержки Java JCA интерфейса.

и функционирует в группах программно-аппаратных сред в соответствии с п.2.

5.3. Состав программного обеспечения

СКЗИ функционирует на уровне приложения. В состав программного обеспечения для всех платформ входят СКЗИ «КриптоПро JCP» версия 2.0 R2 и ПКЗИ.

В состав СКЗИ «КриптоПро JCP» версия 2.0 R2 входят:

- Библиотеки криптопровайдера для исполнений по уровню КС1;
- Библиотеки шифровального провайдера для исполнений по уровню КС1;
- Подсистема контроля целостности;
- Датчик случайных чисел (ДСЧ);
- Биологический датчик случайных чисел для инициализации основного ДСЧ.

В состав ПКЗИ входят следующие компоненты:

- ASN.1 модуль;
- Модуль поддержки ASN.1;
- Модуль запроса сертификатов;
- Подсистема настройки провайдера;
- Модули поддержки считывателей и носителей;

- Java-машина.

5.4. Состав SDK СКЗИ

В состав SDK СКЗИ входят следующие документы, описывающие интерфейсы:

- комплект документации Doc для разработки;
- комплект документации javadoc с описанием классов и модулей.

Так же в состав SDK СКЗИ входят примеры samples.

5.5. Состав СКЗИ

В состав СКЗИ входят следующие компоненты:

- библиотеки провайдеров, реализующие интерфейсы JCA, JCE, JSSE;
- дополнительные библиотеки для реализации взаимодействия с КриптоПро УЦ;
- библиотеки, реализующие стандарты CadES, XadES, XML Dsig;
- библиотеки для интеграции с модулями поддержки и ключевыми носителями;
- ASN.1 библиотека - система кодирования/декодирования данных в форматах ASN.1.

Состав модулей СКЗИ для соответствующих программно-аппаратных сред конкретизируется в дополнениях ЖТЯИ.00091-02 91 01-01. Инструкция по использованию, ЖТЯИ.00091-02 92 01. Правила пользования, ЖТЯИ.00091-02 91 02-01. Инструкция по использованию (JTLS), ЖТЯИ.00091-02 90 01-01. Руководство администратора безопасности, ЖТЯИ.00091-02 33 03-01. Руководство программиста (JTLS), ЖТЯИ.00091-02 33 02-01. Руководство программиста (модули шифрования), ЖТЯИ.00091-02 33 01-01. Руководство программиста.

5.6. Применение СКЗИ

Возможны следующие применения СКЗИ:

- Применение «КриптоПро JCP» версия 2.0 R2 в составе стандартного программного обеспечения Oracle (Sun) и IBM JRE, использующих криптографический интерфейс в соответствии с архитектурой JCA/JCE/JSSE (подробнее см. ЖТЯИ.00091-02 90 01-01. Руководство администратора безопасности. Общая часть);
- Встраивание «КриптоПро JCP» версия 2.0 R2 во вновь разрабатываемое или существующее прикладное программное обеспечение (подробнее см. ЖТЯИ.00091-02 90 01-01. Руководство администратора безопасности. Общая часть и ЖТЯИ.00091-02 33 01-01. Руководство программиста).

6. Использование СКЗИ в стандартном программном обеспечении

Программное обеспечение СКЗИ позволяет использовать российские криптографические алгоритмы и сертификаты открытых ключей стандарта X.509 совместно со следующим программным обеспечением:

- Java-машина J9VM производства IBM «Java(TM) 7 Runtime Environment, Standard Edition» версии 1.7 и «Java(TM) 8 Runtime Environment, Standard Edition» версии 1.8 на 32-битной и 64-битной платформе.
- Java-машина производства Oracle «Java(TM) 7 Runtime Environment, Standard Edition» версии 1.7 и «Java(TM) 8 Runtime Environment, Standard Edition» версии 1.8 на 32-битной и 64-битной платформе.
- Java-машина производства Oracle «Java(TM) 10 Runtime Environment, Standard Edition версии 10 и «Java(TM) 11 Runtime Environment, Standard Edition версии 11 на 64-битной платформе.

•Российские криптографические алгоритмы и сертификаты открытых ключей X.509 используются с указанным программным обеспечением в соответствии со следующими международными и российскими рекомендациями:

•Using the GOST R 34.10-94, GOST R 34.10-2001 and GOST R 34.11-94 algorithms with the Internet X.509 Public Key Infrastructure Certificate and CRL Profile (rfc4491) описывает использование российских криптографических алгоритмов в инфраструктуре открытых ключей интернет (PKIX, Internet X.509 Public Key Infrastructure). В данном документе описаны форматы представления открытых ключей ЭП, используемых для создания сертификатов открытых ключей и списков отозванных сертификатов X.509, идентификаторы алгоритмов, соответствие параметров криптографических алгоритмов их идентификаторам.

•Additional cryptographic algorithms for use with GOST 28147-89, GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 algorithms (rfc4357) описывает дополнительные алгоритмы, необходимые для использования ГОСТ 28147-89, ГОСТ Р 34.10-2001 и ГОСТ Р 34.11-94. В их число входят: блочное шифрование по ГОСТ 28147-89 в режиме сцепления блоков (режиме CBC), режимы дополнения данных для блочного шифрования по ГОСТ 28147-89 в режиме CBC, ключевое хэширование (HMAC на базе ГОСТ Р 34.11-94), преобразование ключа и синхропосылки после обработки очередных 1 Кб данных, генерация псевдослучайной последовательности (аналог PRF на базе HMAC), формирование ключа обмена (согласования) на базе ГОСТ Р 34.10-2001, формирование ключа экспорта рабочего ключа, диверсификация ключа, экспорт рабочего ключа на ключе экспорта, экспорт рабочего ключа на ключе обмена, наборы стандартных параметров алгоритмов (например, для шифрования - узел замены, режим шифрования, алгоритм усложнения ключа), задаваемые идентификаторами.

•Using the GOST 28147-89, GOST R 34.11-94, GOST R 34.10-94 and GOST R 34.10-2001 algorithms with the Cryptographic Message Syntax (CMS) (rfc4490) описывает использование российских криптографических алгоритмов в документах, удовлетворяющих стандарту CMS (Cryptographic Message Syntax), в частности, применяемом для обмена защищёнными сообщениями по электронной почте и

являющимся стандартом представления электронного документа в защищенном виде с использованием электронной подписи и шифрования. Для шифрованных сообщений описаны оба варианта: обмен ключами и транспорт ключа (key agreement и key transport).

•Технический комитет по стандартизации «Криптографическая защита информации» (ТК 26), «Системы обработки информации. Защита криптографическая. Методические рекомендации по криптографическим алгоритмам, сопутствующим применению стандартов ГОСТ Р 34.10-2012 и ГОСТ Р 34.11-2012».

•Технический комитет по стандартизации «Криптографическая защита информации» (ТК 26), «Техническая спецификация. Использование алгоритмов ГОСТ Р 34.10, ГОСТ Р 34.11 в профиле сертификата и списке отзыва сертификатов (CRL) инфраструктуры открытых ключей X.509».

•Технический комитет по стандартизации «Криптографическая защита информации» (ТК 26), «Информационная технология. Криптографическая защита информации. Рекомендации по стандартизации. Использование наборов алгоритмов шифрования на основе ГОСТ 28147-89 для протокола безопасности транспортного уровня (TLS)».

•Технический комитет по стандартизации «Криптографическая защита информации» (ТК 26), «Информационная технология. Криптографическая защита информации. Рекомендации по стандартизации. Использование алгоритмов ГОСТ 28147-89, ГОСТ Р 34.11 и ГОСТ Р 34.10 в криптографических сообщениях формата CMS».

•Технический комитет по стандартизации «Криптографическая защита информации» (ТК 26), «Системы обработки информации. Защита криптографическая. Рекомендации по стандартизации. Задание параметров эллиптических кривых в соответствии с ГОСТ Р 34.10-2012».

•Технический комитет по стандартизации «Криптографическая защита информации» (ТК 26), «Системы обработки информации. Защита криптографическая. Методические рекомендации по заданию узлов замены блока подстановки алгоритма шифрования ГОСТ 28147-89».

7. Встраивание СКЗИ

Архитектура СКЗИ обеспечивает возможность его встраивания в различные программно-аппаратные среды.

8. Использование интерфейсов JCA и JCE

СКЗИ может быть использовано прикладным программным обеспечением через интерфейсы JCA и JCE.

Использование интерфейсов JCA и JCE преследует следующие цели:

- обеспечение доступа к криптографическим функциям на прикладном уровне (генерация ключей, создание/проверка электронной подписи,

шифрование/расшифрование данных, хэширование, взаимодействие с ключевыми контейнерами и сертификатами). Эта цель достигается путем изолирования прикладного уровня от уровня реализации криптографических функций. При этом прикладным программистам не нужно детально изучать особенности реализации того или иного алгоритма или изменять код в зависимости от алгоритма.

- изолирование прикладного уровня от уровня криптографических функций с возможностью использования разных алгоритмов в различных их реализациях, включая аппаратные.

8.1. Функции генерации ключей и работы с ключевыми контейнерами

Данные функции предназначены для генерации ключевой пары на заданном алгоритме с помощью ПДСЧ. Созданная пара может быть в дальнейшем сохранена в ключевой контейнер. Функции работы с ключевыми контейнерами позволяют извлечь ссылку на закрытый ключ и контекст сертификата формата X.509 из контейнера по алиасу.

Классы-генераторы представлены интерфейсом KeyPairGenerator, класс работы с контейнерами — KeyStore, класс закрытого ключа — PrivateKey, класс сертификата и открытого ключа — Certificate и PublicKey.

8.2. Функции хэширования и подписи

Функции данной группы предназначены для хэширования информации, создания подписи и ее проверки.

Класс функции хэширования представлен интерфейсом MessageDigest, класс подписи — Signature.

8.3. Функции шифрования и согласования ключей

Классы группы шифрования предназначены для зашифрования и расшифрования данных, а также экспорта и импорта секретных ключей. Классы согласования ключей предназначены для выработки ключа согласования на основе ключей отправителя и получателя.

Класс шифрования представлен интерфейсом Cipher, класс согласования ключей — KeyAgreement.

8.4. Функции кодирования/декодирования

Данные функции предназначены для преобразования (кодирования) из внутреннего представления объектов, во внешнее представление и обратно. В качестве внешнего представления объектов используется формат ASN.1 (Abstracy Syntax Notation One), определенный серией рекомендаций X.680. К этой же группе функций может быть отнесен набор функций, позволяющих расширить функциональность JCA/JCE путем реализации и регистрации собственных типов объектов.

8.5. Высокоуровневые функции обработки криптографических сообщений формата формата CMS, CAdES, XAdES

Эта группа функций в первую очередь предназначена для использования в прикладном программном обеспечении. С их помощью можно:

- зашифровать/расшифровать сообщения от одного пользователя к другому;
- подписать данные подписью формата CAdES, XAdES;
- проверить подпись данных формата CMS, CAdES, XAdES.

Эти функции (как и функции низкого уровня) оперируют сертификатами открытых ключей X.509 для адресации отправителя/получателя данных. В качестве формата данных используется формат PKCS#7 или CMS, CAdES или XAdES.

СКЗИ «КриптоПро JCP» версия 2.0 R2 использует классы фабрик сертификатов встроенных провайдеров типа Oracle (Sun) или IBM и их реализации классов сертификатов и CRL, поддерживает сертификаты открытых ключей стандарта X.509v3 согласно RFC 5280 «Internet X.509 Public Key Infrastructure. Certificate and Certificate Revocation List (CRL) Profile» с учетом RFC 4491 «Using the GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 Algorithms with the Internet X.509 Public Key Infrastructure Certificate and CRL Profile», а также документа Технического комитета по стандартизации «Криптографическая защита информации» (ТК 26), «Техническая спецификация. Использование алгоритмов ГОСТ Р 34.10, ГОСТ Р 34.11 в профиле сертификата и списке отзыва сертификатов (CRL) инфраструктуры открытых ключей X.509».

СКЗИ «КриптоПро JCP» версия 2.0 R2 поддерживает формат криптографических сообщений согласно RFC 3852 «Cryptographic Message Syntax (CMS)» с учетом RFC 4490 «Using the GOST 28147-89, GOST R 34.11-94, GOST R 34.10-94, and GOST R 34.10-2001 Algorithms with Cryptographic Message Syntax (CMS)», а также документа Технического комитета по стандартизации «Криптографическая защита информации» (ТК 26), «Информационная технология. Криптографическая защита информации. Рекомендации по стандартизации. Использование алгоритмов ГОСТ 28147-89, ГОСТ Р 34.11 и ГОСТ Р 34.10 в криптографических сообщениях формата CMS».

8.6. Функции формирования запроса на сертификат

Данные функции предназначены для создания запроса на сертификат. В запрос формата PKCS10 могут быть включены поля: субъект, издатель, дата начала и дата окончания срока действия сертификата, использование ключа, открытый ключ, дополнительные расширения. Запрос подписывается закрытым ключом.

С помощью функций работы с УЦ версии 1.3-1.5 или 2.0 запрос может быть передан в формате DER или BASE64 в соответствующий УЦ, после чего может быть получен сертификат.

Класс для формирования запроса на сертификат — GostCertificateRequest, CA15-GostCertificateRequest или CA20GostCertificateRequest. В составе модуля для формирования запроса есть необходимый функционал для передачи запроса по защищенному каналу в УЦ.

9. Использование интерфейса PKIX

СКЗИ позволяет прикладному программному обеспечению использовать интерфейс PKIX.

Использование интерфейсов PKIX преследует следующие цели:

- обеспечение доступа к функциям построения и проверки цепочки сертификтов (X.509) на отзыв на прикладном уровне. Эта цель достигается путем изолирования прикладного уровня от уровня реализации функций PKIX. При этом прикладным программистам не нужно детально изучать особенности реализации того или иного алгоритма или изменять код в зависимости от алгоритма.

Алгоритмы модуля проверки цепочки сертификатов расширяют стандартный PKIX, поставляемый провайдером Sun (Oracle) или IBMJCE по умолчанию.

Класс для построения цепочки сертификатов представлен интерфейсом CertPath-Builder, класс проверки цепочки сертификатов — CertPathValidator.

Процедура построения цепочки сертификатов учитывает ряд проверок: проверка срока действия сертификата, проверка назначения, подчиненности сертификатов, проверка расширений и т. д. Процедура проверки цепочки сертификатов выполняется путем обращения к OCSP службе или CRL DP, или CRL в виде файлов.

10. Поддержка протокола TLS

Модуль JTLS (cpSSL) позволяет реализовать защищенный сетевой протокол в соответствии с рекомендациями RFC 2246 «The TLS Protocol. Version 1.0» и «Технический комитет по стандартизации «Криптографическая защита информации» (ТК 26), «Информационная технология. Криптографическая защита информации. Рекомендации по стандартизации. Использование наборов алгоритмов шифрования на основе ГОСТ 28147-89 для протокола безопасности транспортного уровня (TLS)». Модуль обеспечивает двустороннюю и одностороннюю аутентификацию приложений при их взаимодействии по сети с использованием алгоритма ЭП и сертификатов открытых ключей, а также шифрование данных, передаваемых в сетевом соединении.

Прикладное программное обеспечение может использовать протокол TLS для аутентификации и защиты данных, передаваемых по собственным протоколам на основе TCP/IP и HTTPS.

Протокол TLS (Transport Layer Security, спецификация IETF - RFC2246) относится к средствам защиты прикладных пакетов Microsoft Internet Explorer, Internet Information Services (IIS), Microsoft SQL Server 2000 и COM+. Он обеспечивает аутентификацию связывающихся сторон, конфиденциальность и целостность пересылаемой информации. Аутентификация обеспечивается использованием сертификатов стандарта X.509 (в средах с сильной аутентификацией), конфиденциальность – шифрованием пересылаемых данных, целостность — применением хэш-функции и кода аутентификации сообщения (Message Authenticity Code, MAC).

Для подключения по протоколу TLS используется префикс https, при этом обозреватель Web-сервера по умолчанию будет подключаться к порту TCP 443 вместо

стандартного порта TCP 80. Если сервер не поддерживает протокол TLS, соединение не устанавливается. Применение протоколов SSL/TLS (SSL - более ранние версии протокола) показано в Таблице 1.

Таблица 1 - Применение протокола SSL/TLS

Протокол	Порт	Описание
HTTPS	443	HTTP по SSL/TLS
SMTPS	465	SMTP (электронная почта) по SSL/TLS
NNTPS	563	NNTP (новости) по SSL/TLS
LDAPS	636	LDAP (доступ к каталогам) по SSL/TLS
POP3S	995	POP (электронная почта) по SSL/TLS
IRCS	994	IRC по SSL/TLS
IMAPS	993	IMAP (электронная почта) по SSL/TLS
FTPS	990	FTP (передача файлов) по SSL/TLS

Для того, чтобы протокол SSL/TLS действовал, Web-сервер должен иметь пару сертификат открытого ключа/закрытый ключ. Владелец сертификата должен подтвердить, что он является владельцем закрытого ключа, связанного с сертификатом. Это дает возможность клиенту аутентифицировать сервер, с которым он хочет связаться.

В процессе взаимной аутентификации:

- выполняется криптографическая проверка наличия у сервера закрытого ключа, соответствующего открытому ключу, указанному в сертификате;
- проверяется степень доверия издателю сертификата;
- проверяется, не истек ли срок действия сертификата;
- проверяется, не отозван ли сертификат; по умолчанию Internet Explorer эту проверку не выполняет — это делает IIS.

Если любая из указанных проверок приводит к отрицательному результату, пользователь получает предупреждение и может разорвать соединение (это рекомендуется сделать).

Достигнув доверия, стороны вырабатывают сеансовый ключ, на основе которого обеспечивается шифрование данных в течение сеанса.

10.1. Основные понятия протокола TLS

Протокол TLS предназначен для обеспечения криптографическими средствами аутентификации отправителя (клиента) и адресата (сервера), контроля целостности и шифрования данных информационного обмена.

Аутентификация опционально может быть односторонней (аутентификация сервера клиентом), взаимной (встречная аутентификация сервера и клиента) или не использоваться.

Иерархия информационного обмена включает в себя сессии, соединения и поток сообщений в соединении. Поток сообщений при большой длине разбивается на фрагменты с пакетной передачей фрагментов. В одной сессии может быть реализовано несколько соединений, произвольно разнесенных по времени. В каждом соединении может быть обработан необходимый поток сообщений.

Сессия характеризуется следующими атрибутами:

- идентификатор сессии (случайное число, 32 байта, задается сервером при открытии сессии);
- метод компрессии;
- сертификат сервера (опционально);
- сертификат клиента (опционально);
- спецификация алгоритмов и параметров защиты (алгоритмы шифрования и MAC, криптографические параметры);
- master secret (используется при генерации ключей шифрования, ключей MAC, векторов инициализации);
- флаг, разрешающий/запрещающий новые соединения в сеансе.

Сертификаты представляются в стандарте X509. v3. Спецификация алгоритмов и параметров защиты может меняться в течение сессии.

Соединение характеризуется следующими атрибутами:

- client_random – случайные 32 байта, задаваемые клиентом;
- server_random– случайные 32 байта, задаваемые сервером;
- client write MAC secret (ключ клиента для вычисления значения ключевой хэш-функции);
- server write MAC secret (ключ сервера для вычисления значения ключевой хэш-функции);
- client write key (ключ, используемый для шифрования данных клиентом и расшифрования их сервером);
- server write key (ключ, используемый для шифрования данных сервером и расшифрования их клиентом);
- client write IV, server write IV (векторы инициализации, используемые клиентом и сервером соответственно);
- порядковый номер соединения (поддерживается независимо для передаваемых и принимаемых сообщений).

Вектор инициализации задается для первого фрагмента сообщения в соединении; для последующих фрагментов вектор инициализации формируется из конечного блока зашифрованного текста предыдущего фрагмента.

Порядковые номера соединений поддерживаются независимо для передаваемых и принимаемых сообщений. При смене сессии, изменении спецификации алгоритмов и параметров защиты нумерация соединений начинается с 0; диапазон нумерации: $0 \div 2^{64}-1$.

Соединение ассоциируется с одной сессией.

Алгоритм преобразования информации при обмене с использованием протокола TLS включает следующие операции:

- прием от протокола верхнего уровня потока не интерпретируемых данных в блоках произвольного размера;
- фрагментация принятых с верхнего уровня данных в структурированные блоки (фрагменты) протокола TLS. Размер фрагмента – не более 2^{14} байт;
- компрессия фрагментов (опционально);
- вычисление значения ключевой хэш-функции (MAC) от конкатенации ключа хэш-функции, типа компрессии, длины компрессированного фрагмента, компрессированного фрагмента и заданной константы;
- конкатенация фрагмента и результата вычисления значения хэш-функции от него (расширенный фрагмент);
- зашифрование расширенного фрагмента (опционально);
- добавление открытого заголовка, содержащего тип сообщения (один байт), версию протокола TLS (два байта) и длину компрессированного фрагмента.

При приеме информации применяется обратная последовательность операций.

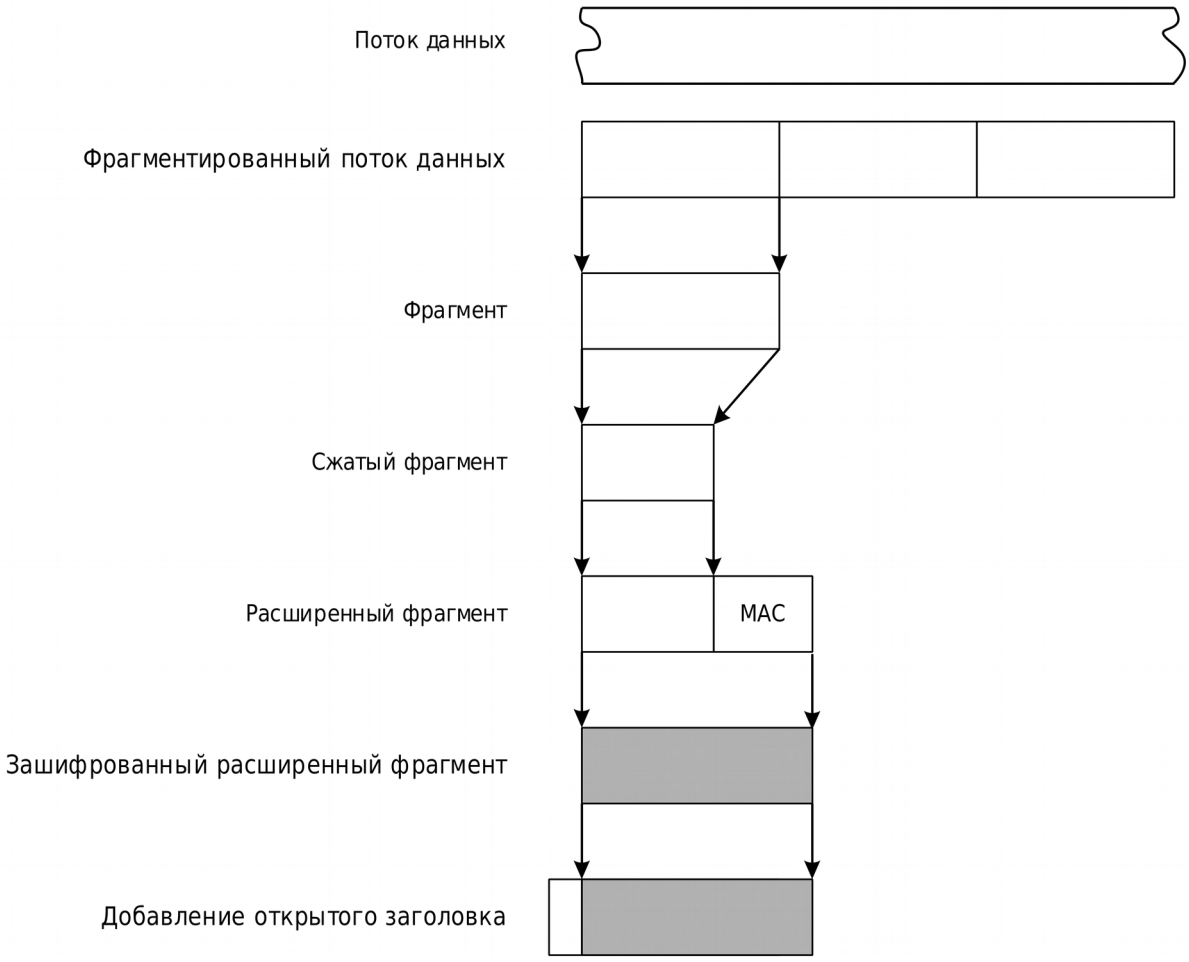

Рисунок 2 — Алгоритм преобразования информации при обмене с использованием протокола TLS

В протоколе TLS используются следующие типы сообщений:

- Hello message (ClientHello, ServerHello);
- Change cipher specs message (изменение спецификации алгоритмов и параметров защиты);
- Key exchange message (передача ключа обмена ключами шифрования и MAC клиента, сервера);
- Alert message (предупреждение, оповещение о фатальной ошибке);
- Application_data message (передача данных);
- Finished message (сообщение о возможности работы в созданной сессии).

Протокол TLS является двухуровневым и действует над транспортным протоколом. К первому уровню относятся TLS Handshake Protocol, TLS Change Cipher Spec и TLS Alert Protocol. Ко второму уровню относится TLS Record Protocol.

TLS Handshake Protocol обеспечивает инициализацию сессии (соединения) выполнением следующих операций:

- клиент и сервер договариваются об используемых в сессии алгоритмах и параметрах защиты, обмениваются случайными величинами client_random, server_random, договариваются, будут или нет новые соединения;
- производится обмен сертификатами для аутентификации клиента и сервера (по заданным опциям);
- клиент генерирует случайную величину pre_master secret, шифрует ее и передает серверу.
- клиент и сервер по pre_master secret, client_random и server_random формируют master secret (набор необходимой ключевой информации) сессии.

TLS Handshake Protocol работает по следующей схеме, представленной на Рисунке 3.

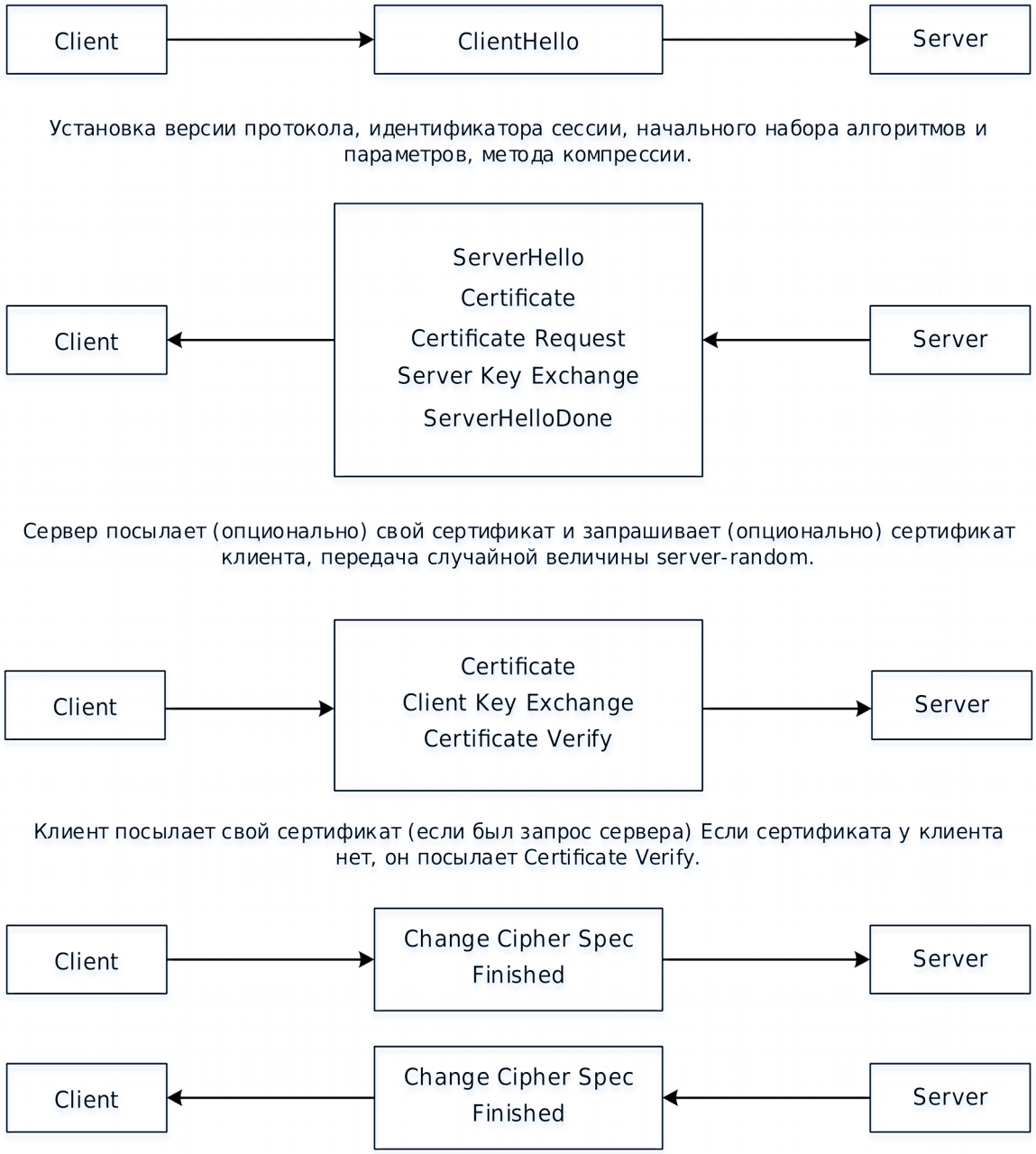

Рисунок 3 – Схема работы TLS Handshake Protocol

10.2. Модуль сетевой аутентификации «КриптоПро JTLS»

Модуль сетевой аутентификации «КриптоПро JTLS» (cpSSL) реализован на базе протокола TLS v.1.0 и российских стандартов криптографической защиты конфиденциальной информации (алгоритмы шифрования в соответствии с ГОСТ 28147-

89, алгоритмы выработки и проверки электронной подписи в соответствии с ГОСТ Р 34.10-2001 и ГОСТ Р 34.10-2012, алгоритмы хэширования в соответствии с ГОСТ Р 34.11-94 и ГОСТ Р 34.11-2012). Используется также алгоритм Диффи-Хеллмана открытого распределения ключей на базе ГОСТ Р 34.10-2001 и ГОСТ Р 34.10-2012.

Аутентификация клиент-сервер может быть односторонней и двусторонней.

Односторонняя аутентификация обеспечивает минимально необходимый уровень защиты, и включает в себя:

- обязательную аутентификацию сервера без аутентификации клиентов;
- шифрование трафика между клиентом и сервером.

При работе в данном режиме сервер на этапе «рукопожатия» не запрашивает сертификат клиента и устанавливается «анонимное» защищенное соединение. В этом случае клиент может не иметь закрытого ключа и сертификата, однако при этом он лишается возможности формировать электронную подпись под документами. Режим с односторонней аутентификацией сервера может использоваться для предоставления некоторой группе пользователей конфиденциальной информации на основании парольной защиты, однако пароль в этом случае будет предъявляться пользователем только после установления защищенного TLS-соединения с Web-сервером, что повышает уровень защиты от несанкционированного доступа по сравнению с передачей пароля по открытым соединениям. При односторонней аутентификации сервер запрашивает сертификат клиента, но его отсутствие не считается ошибкой.

Двусторонняя аутентификация включает в себя:

- взаимную аутентификацию клиента и Web-сервера с помощью их сертификатов;
- шифрование трафика между клиентом и сервером;
- формирование и проверку электронной подписи под электронными HTML-формами, заполняемыми пользователями.

Двусторонняя аутентификация позволяет обеспечить доступ в закрытую часть Web-сервера только зарегистрированным владельцам сертификатов. При этом нужно иметь в виду, что разграничение доступа к информационным ресурсам сервера, основанное на проверке сертификатов клиентов, гораздо надежнее, чем просто парольная защита.

В данном режиме работы клиенту необходимо сгенерировать закрытый и открытый ключи и получить сертификат открытого ключа в УЦ.

11. Примеры использования СКЗИ «КриптоПро JCP» версия 2.0 R2

Для разработчиков в состав дистрибутива СКЗИ «КриптоПро JCP» версия 2.0 R2 включается javadoc-документация, содержащая описание реализуемых интерфейсов и примеры использования на уровне вызова основных функций JCA/JCE в модуле samples-sources.jar. В состав дистрибутива включены также примеры использования «КриптоПро JCP» версия 2.0 R2 для подписи/проверки подписи XML, использования CAdES, XAdES,

примеры создания запросов на сертификаты и взаимодействия с УЦ, примеры осуществления взаимодействия по защищенному протоколу (TLS).

На форуме Крипто-Про (http://www.cryptopro.ru/CryptoPro/forum2/) ведется обсуждение по вопросам использования криптографических функций и сертификатов открытых ключей и ключей проверки ЭП.

Все вышеперечисленные варианты встраивания и использования СКЗИ «КриптоПро JCP» версия 2.0 R2 должны применяться с учетом п. 1.8 Формуляра. При этом указанные в настоящем документе интерфейсы являются уровнями встраивания СКЗИ «КриптоПро JCP» версия 2.0 R2 в прикладные системы и не являются приложениями, входящими в состав операционных систем.

12. История версий

12.1. «КриптоПро JCP» версия 1.0.54

- Включено исполнение класса защиты КС1, функционирующее на программно-аппаратных платформах в соответствии с п.2;
- Добавлен модуль CAdES для создания и проверки усовершенствованной подписи;
- Добавлен функционал для взаимодействия с КриптоПро УЦ 1.5.

12.2. «КриптоПро JCP» версия 1.0.55

- Доработан модуль сетевой аутентификации JTLS – добавлена процедура проверки цепочки сертификатов.

12.3. «КриптоПро JCP» версия 2.0 R2

- Реализована поддержка алгоритмов формирования и проверки электронной подписи по ГОСТ Р 34.10-2012;
- Реализована поддержка алгоритмов хэширования по ГОСТ Р 34.11-2012;
- Реализована поддержка алгоритмов, сопутствующих применению стандартов ГОСТ Р 34.10-2012 и ГОСТ Р 34.11-2012;
- Реализована поддержка SSLEngine в модуле сетевой аутентификации;
- Реализована поддержка renegotiation indication extension в модуле сетевой аутентификации:
- Доработан БиоДСЧ;
- Доработан провайдер для поддержки Java версии выше 1.7.0_25;
- Для алгоритма ГОСТ Р 34.10-2012 реализована возможность вычисления кратной точки на основе эллиптических кривых в форме Эдвардса;
- Добавлена поддержка КриптоПро УЦ версии 1.5 и 2.0;

- Добавлена поддержка создания, усовершенствования и проверки подписи формата CadES (BES, T, X-Long Type 1);
- Добавлена поддержка зашифрования и расширования в формате Enveloped CMS (enveloped-data);
- Добавлена поддержка создания и проверки подписи формата XAdES (BES, T, X Long Type 1);
- Выполнена интеграция модуля сетевой аутентификации с веб-серверами: glass-fish, jboss, IBM WebSphere, jetty;
- Добавлен установщик для ОС Windows;
- Добавлен собственный модуль работы с Rutoken;
- Добавление расширения «срок действия закрытого ключа» при создании ключевого контейнера;
- Поддержка CadES-A подписи.

13. Информация для пользователей

Для получения дополнительной информации о данном продукте, а также о других продуктах ООО «КРИПТО-ПРО», можно обращаться по адресу:

Служба маркетинга и технической поддержки Крипто-Про.

127018, Москва, Сущевский вал 18, ООО «КРИПТО-ПРО».

Телефон: +7 (495) 995 4820

Факс: +7 (495) 995 4820

e-mail: info@CryptoPro.ru WWW: http://www.CryptoPro.ru