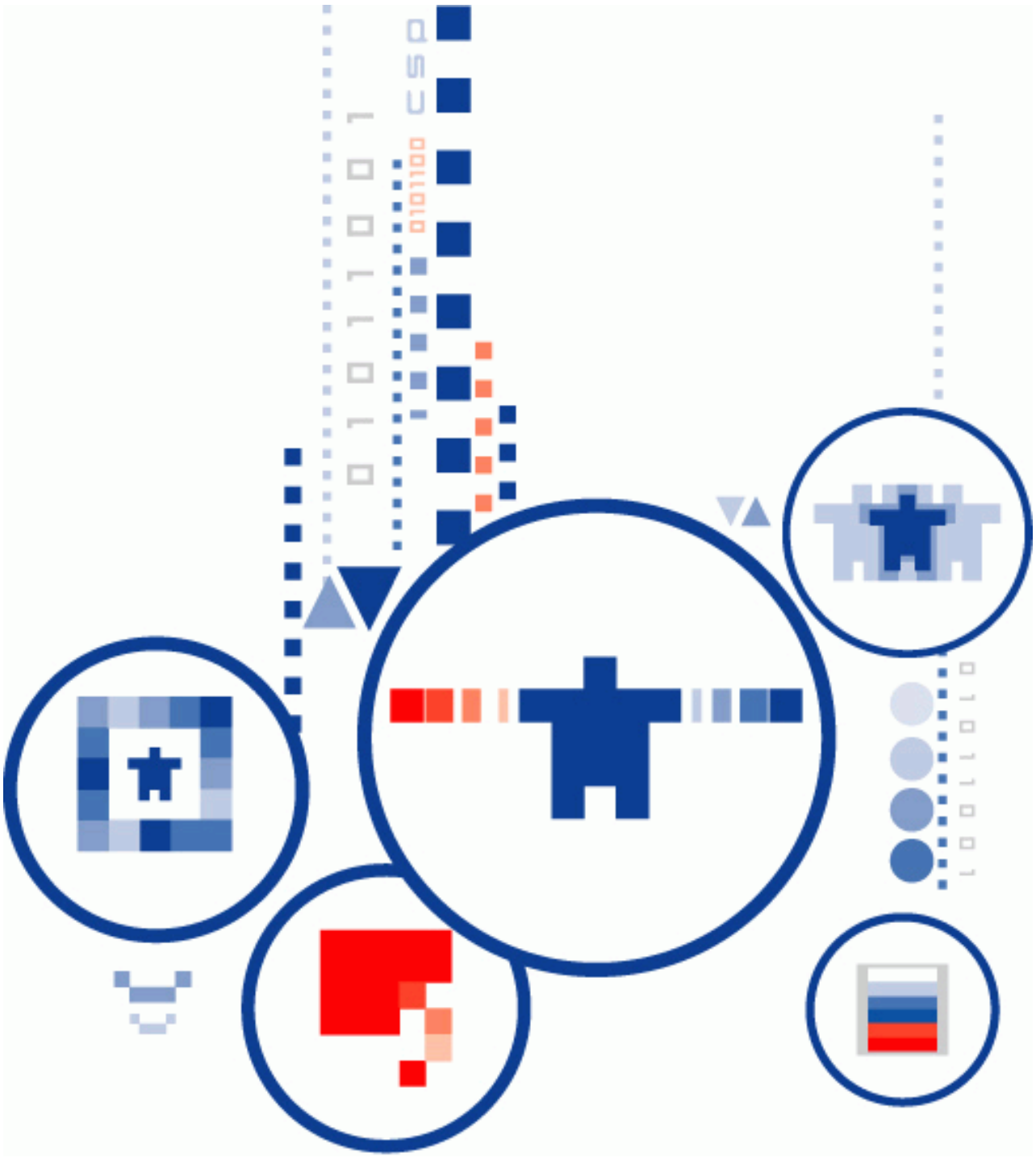
УТВЕРЖДЁН
ЖТЯИ.00094-01 94 03



Службы УЦ версии 2.0. Служба проверки сертификатов и электронной подписи

КриптоПро SVS

Руководство разработчика

АННОТАЦИЯ

Настоящий документ содержит описание программного интерфейса сервиса проверки подписи «КриптоПро SVS 2.0», предназначенной для установления статуса сертификата ключа проверки электронной подписи и выполнения процедуры подтверждения подлинности электронных подписей документов различного формата.

Документ предназначен для разработчиков как руководство по использованию программного интерфейса «КриптоПро SVS» версии 2.0.xxx.

Информация о разработчике «КриптоПро SVS»:

ООО «Крипто-Про»
127018, Москва, ул. Сущёвский вал, 18
Телефон: (495) 995 4820
Факс: (495) 995 4820
http://www.CryptoPro.ru
E-mail: info@CryptoPro.ru

Лист истории изменений

Версия	**Описание изменений**
2.0.1777	Добавлено описание методов проверки подписи, возвращающие расширенную информацию (раздел 1.1.1, 1.1.4, 1.1.6, 1.1.8, 1.4.1, 1.5.3).
2.0.1917	Добавлено описание метода для проверки необработанной подписи (раздел 1.1.10). Расширено описание структуры VerifyParams (см. раздел 1.5.4).
2.0.xxx	Добавлено описание интерфейса REST для работы с Сервисом Проверки Подписи.

СОДЕРЖАНИЕ

1. Описание интерфейса SOAP для работы с сервисом проверки подписи «КриптоПро SVS»

Веб-сервис доступен по адресу:

- http://<hostname>/<ApplicationName>/service.svc

Описание веб-сервиса доступно по адресу:

- http://<hostname>/<ApplicationName>/service.svc?wsdl

где <ApplicationName> – имя веб-приложения Сервиса Проверки Подписи, по умолчанию имеет значение VerificationService.

Описание интерфейса сервиса находится в библиотеке CryptoPro.DSS.Common.dll, имя интерфейса CryptoPro.DSS.Common.Service.IVerificationService.

Интерфейс IVerificationService предоставляет следующие методы:

- VerifySignature
- VerifySignatureEx
- VerifySignatureAll
- VerifyDetachedSignature
- VerifyDetachedSignatureEx
- VerifyDetachedSignatureAll
- VerifyDetachedSignatureAllEx
- VerifyGost34102001
- VerifyRawSignature
- VerifyCertificate
- GetSignersInfo
- GetPolicy

1.1. Функции для проверки подписи

SOAP-интерфейс для работы с сервисом проверки подписи «КриптоПро SVS» предоставляет следующие функции для проверки подписи:

- VerifySignature
- VerifySignatureEx
- VerifySignatureAll
- VerifySignatureAllEx
- VerifyDetachedSignature
- VerifyDetachedSignatureEx
- VerifyDetachedSignatureAll
- VerifyDetachedSignatureAllEx
- VerifyGost34102001
- VerifyRawSignature

1.1.1. Функция VerifySignatureEx

Функция **VerifySignatureEx** проверяет присоединенную подпись.

```
VerificationResultEx VerifySignatureEx(SignatureType signatureType, byte[] document,
Dictionary<VerifyParams, string> verifyParams)
```

Параметры:

- **signatureType** – формат подписи SignatureType,
- **document** – документ с подписью в виде массива байт,

- **verifyParams** – словарь дополнительных параметров проверки подписи (опциональный параметр). Типы дополнительных параметров указаны в перечислении VerifyParams и разделе 1.6.

Возвращаемое значение: VerificationResultEx.

Для форматов подписи CMS и CAdES подпись может быть передана в виде массива байт, так и в кодировке Base64. Если подпись была передана в кодировке Base64, она автоматически будет декодирована в массив байт для проверки.

1.1.2. Функция VerifySignature

Функция **VerifySignature** проверяет присоединенную подпись.

```
VerificationResult VerifySignature(SignatureType signatureType, byte[] document,
Dictionary<VerifyParams, string> verifyParams)
```

Параметры:

- **signatureType** – формат подписи SignatureType,
- **document** – документ с подписью в виде массива байт,
- **verifyParams** – словарь дополнительных параметров проверки подписи (опциональный параметр). Типы дополнительных параметров указаны в перечислении VerifyParams и разделе 1.6.

Возвращаемое значение: VerificationResult.

Для форматов подписи CMS и CAdES подпись может быть передана в виде массива байт, так и в кодировке Base64. Если подпись была передана в кодировке Base64, она автоматически будет декодирована в массив байт для проверки.

1.1.3. Функция VerifySignatureAll

Функция **VerifySignatureAll** проверяет присоединенную подпись.

```
IList<VerificationResult> VerifySignatureAll(SignatureType signatureType, byte[]
document, Dictionary<VerifyParams, string> verifyParams)
```

Параметры:

- **signatureType** – формат подписи SignatureType,
- **document** – документ с подписью в виде массива байт,
- **verifyParams** – словарь дополнительных параметров проверки подписи (опциональный параметр). Типы дополнительных параметров указаны в перечислении VerifyParams. Параметр не используется и зарезервирован для будущего использования. В качестве значения передавать null.

Возвращаемое значение: IList<VerificationResult>.

Для форматов подписи CMS и CAdES подпись может быть передана в виде массива байт, так и в кодировке Base64. Если подпись была передана в кодировке Base64, она автоматически будет декодирована в массив байт для проверки.

1.1.4. Функция VerifySignatureAllEx

Функция **VerifySignatureAllEx** проверяет присоединенную подпись.

```
IList<VerificationResultEx> VerifySignatureAllEx(SignatureType signatureType, byte[]
document, Dictionary<VerifyParams, string> verifyParams)
```

Параметры:

- **signatureType** – формат подписи SignatureType,
- **document** – документ с подписью в виде массива байт,
- **verifyParams** – словарь дополнительных параметров проверки подписи (опциональный параметр). Типы дополнительных параметров указаны в перечислении VerifyParams. Параметр не используется и зарезервирован для будущего использования. В качестве значения передавать null.

Возвращаемое значение: IList<VerificationResultEx>.

Для форматов подписи CMS и CAdES подпись может быть передана в виде массива байт, так и в кодировке Base64. Если подпись была передана в кодировке Base64, она автоматически будет декодирована в массив байт для проверки.

1.1.5. Функция VerifyDetachedSignature

Функция **VerifyDetachedSignature** проверяет откреплённую подпись.

```
VerificationResult VerifyDetachedSignature(SignatureType signatureType, byte[]
document, byte[] signature, Dictionary<VerifyParams, string> verifyParams)
```

Параметры:

- **signatureType** – формат подписи SignatureType,
- **document** – документ в виде массива байт,
- **signature** – подпись в виде массива байт,
- **verifyParams** – словарь дополнительных параметров проверки подписи (опциональный параметр). Типы дополнительных параметров указаны в перечислении VerifyParams и разделе 1.6.

Возвращаемое значение: VerificationResult.

Допустимые значения параметра signatureType: CMS, CAdES (см. примечание к перечислению SignatureType)

1.1.6. Функция VerifyDetachedSignatureEx

Функция **VerifyDetachedSignatureEx** проверяет откреплённую подпись.

```
VerificationResultEx VerifyDetachedSignature(SignatureType signatureType, byte[]
document, byte[] signature, Dictionary<VerifyParams, string> verifyParams)
```

Параметры:

- **signatureType** – формат подписи SignatureType,
- **document** – документ в виде массива байт,
- **signature** – подпись в виде массива байт,
- **verifyParams** – словарь дополнительных параметров проверки подписи (опциональный параметр). Типы дополнительных параметров указаны в перечислении VerifyParams и разделе 1.6.

Возвращаемое значение: VerificationResultEx.

Допустимые значения параметра signatureType: CMS, CAdES (см. примечание к перечислению SignatureType)

1.1.7. Функция VerifyDetachedSignatureAll

Функция **VerifyDetachedSignatureAll** проверяет откреплённую подпись.

```
IList<VerificationResult> VerifyDetachedSignatureAll(SignatureType signatureType,
byte[] document, byte[] signature, Dictionary<VerifyParams, string> verifyParams)
```

Параметры:

- **signatureType** – формат подписи SignatureType,
- **document** – документ в виде массива байт,
- **signature** – подпись в виде массива байт,
- **verifyParams** – словарь дополнительных параметров проверки подписи (опциональный параметр). Типы дополнительных параметров указаны в перечислении VerifyParams. Параметр не используется и зарезервирован для будущего использования. В качестве значения передавать null.

Возвращаемое значение: IList<VerificationResult>.

Допустимые значения параметра signatureType: CMS, CAdES (см. примечание к перечислению SignatureType)

1.1.8. Функция VerifyDetachedSignatureAllEx

Функция **VerifyDetachedSignatureAllEx** проверяет откреплённую подпись.

```
IList<VerificationResultEx> VerifyDetachedSignatureAllEx(SignatureType signatureType,
byte[] document, byte[] signature, Dictionary<VerifyParams, string> verifyParams)
```

Параметры:

- **signatureType** – формат подписи SignatureType,
- **document** – документ в виде массива байт,
- **signature** – подпись в виде массива байт,
- **verifyParams** – словарь дополнительных параметров проверки подписи (опциональный параметр). Типы дополнительных параметров указаны в перечислении VerifyParams. Параметр не используется и зарезервирован для будущего использования. В качестве значения передавать null.

Возвращаемое значение: IList<VerificationResultEx>.

Допустимые значения параметра signatureType: CMS, CAdES (см. примечание к перечислению SignatureType)

1.1.9. Функция VerifyGost34102001

Функция **VerifyGost34102001** проверяет необработанную подпись ГОСТ Р 34.10-2001.

```
VerificationResult VerifyGost34102001(byte[] certificate, byte[] signature, byte[]
data, Dictionary<VerifyParams, string> verifyParams)
```

Параметры:

- **certificate** – сертификат ключа проверки подписи,
- **signature** – подпись в виде массива байт,
- **data** – подписанные данные в виде массива байт,

- **verifyParams** – словарь дополнительных параметров проверки подписи (опциональный параметр). Типы дополнительных параметров указаны в перечислении VerifyParams.

Возвращаемое значение: VerificationResult.

1.1.10. Функция VerifyRawSignature

Функция VerifyRawSignature проверяет необработанную подпись ГОСТ Р 34.10-2001, ГОСТ Р 34.10-2012, RSA и т.п.

```
VerificationResult VerifyRawSignature(byte[] certificate, byte[] signature, byte[] data, Dictionary<VerifyParams, string> verifyParams)
```

Параметры:

- **certificate** – сертификат ключа проверки подписи,
- **signature** – подпись в виде массива байт,
- **data** – подписанные данные в виде массива байт,
- **verifyParams** – словарь дополнительных параметров проверки подписи. Типы дополнительных параметров указаны в перечислении VerifyParams.

Возвращаемое значение: VerificationResult.

В словаре дополнительных параметром необходимо задать параметр HashAlgorithm. Список допустимых значений параметра приведён в **Таблица**

1.2. Функции для проверки сертификата

1.2.1. Функция VerifyCertificate

Функция **VerifyCertificate** проверяет действительность сертификата.

```
VerificationResult VerifyCertificate(byte[] certificate)
```

Параметры:

- **certificate** – сертификат в виде массива байт.

Возвращаемое значение: VerificationResult.

Сертификат может быть передан как в виде массива байт, так и в кодировке Base64 с/без заголовков. Если сертификат был передан в кодировке Base64, он автоматически будет декодирован в массив байт для проверки.

1.3. Другие функции

1.3.1. Функция GetSignersInfo

Функция GetSignersInfo возвращает информацию о найденных в документе подписях.

```
SignersInfo GetSignersInfo(SignatureType signatureType, byte[] document)
```

Параметры:

- **signatureType** – формат подписи SignatureType,
- **document** – подписанный документ.

Возвращаемое значение: SignersInfo.

1.3.2. Функция GetPolicy

Функция GetPolicy возвращает настройки Сервиса Проверки Подписи.

```
VsPolicy GetPolicy()
```

Возвращаемое значение: VsPolicy.

1.4. Типы данных

1.4.1. SignatureTypeDescription

Класс содержит описание поддерживаемых форматов подписи.

Таблица 1. Поля класса SignatureTypeDescription

Имя	Тип	Описание
SignatureType	SignatureType	Перечисление, содержащее возможные форматы подписи
FileExtensions	IList<string>	Связанные расширения файлов

1.4.2. SignersInfo

Класс описывает информацию о найденных в документе подписях. Описание полей класса приведено в **Таблица 2**.

Таблица 2. Описание полей класса SignersInfo

Имя	Тип	Описание
SignerInfoList	IList<SignerInfo>	Информация о найденных подписях

1.4.3. SignerInfo

Класс описывает информацию о подписи в документе. Описание полей класса приведено в **Таблица 3**.

Таблица 3. Описание полей класса SignerInfo

Имя	Тип	Описание
Id	string	Идентификатор узла подписи.
ParentId	string	Идентификатор родительского узла подписи.
Index	int	Порядковый номер узла подписи. Порядковый номер начинается с 1.
SignerCertificateInfo	Dictionary<CertificateInfoParams, string>	Отображаемые данные о сертификате.

1.4.4. VsPolicy

Класс описывает настройки Сервиса Проверки Подписи. Описание полей класса приведено в **Таблица 4**.

Таблица 4. Описание полей класса VsPolicy

Имя	Тип	Описание
AllowedSignatureTypes	IList<SignatureType>	Список форматов подписи, доступных для проверки.
SignatureDescriptions	IList<SignatureTypeDescription>	

1.4.5. VerificationResult

Класс описывает результат проверки подписи или сертификата. Описание полей класса приведено в **Таблица 5**.

Таблица 5. Описание полей класса VerificationResult

Имя	Тип	Описание
Message	string	Суммарная информация о результатах проверки подписи.
Result	bool	Результат проверки.
SignerCertificate	byte[]	Сертификат подписи.
SignerCertificateInfo	Dictionary<CertificateInfoParams, string>	Набор сведений о сертификате подписи.

1.4.1. VerificationResultEx

Класс описывает результат проверки подписи или сертификата. Описание полей класса приведено в **Таблица 5**.

Таблица 6. Описание полей класса VerificationResultEx

Имя	Тип	Описание
Message	string	Суммарная информация о результатах проверки подписи.
Result	bool	Результат проверки.
SignerCertificate	byte[]	Сертификат подписи.
SignerCertificateInfo	Dictionary<CertificateInfoParams, string>	Набор сведений о сертификате подписи.
SignatureInfo	Dictionary<SignatureInfoParams, string>	Набор сведений о подписи.

1.5. Перечислимые типы данных

1.5.1. SignatureType

Перечисление содержит возможные форматы подписи.

Таблица 7. Поля перечисления SignatureType

Имя	Описание
XMLDSig	Подпись документа в формате XMLDSig
GOST3410	Электронная подпись по ГОСТ Р 34.10– 2001 и ГОСТ Р 34.10– 2012
CAdES	Подпись формата CAdES
PDF	Подпись PDF документов
MSOffice	Подпись документов MS Word и Excel
CMS	Подпись формата CAdES-BES

В рамках Сервиса Проверки Подписи значения CAdES и CMS являются эквивалентными и взаимозаменяемыми.

1.5.2. CertificateInfoParams

Перечисление содержит возможные значения отображаемых данных о сертификате.

Таблица 8. Поля перечисления CertificateInfoParams

Имя	Описание
SubjectName	X500 имя субъекта сертификата
IssuerName	X500 имя издателя сертификата
NotAfter	Окончание срока действия сертификата
NotBefore	Начало срока действия сертификата
SerialNumber	Серийный номер сертификата
Thumbprint	Отпечаток сертификата

1.5.3. SignatureInfoParams

Перечисление содержит возможные значения отображаемых данных о сертификате.

Таблица 9. Поля перечисления SignatureInfoParams

Имя	Описание
CAdESType	Формат подписи CAdES. Возможные значения: BES, T, XLT1
SigningTime	Время (UTC) подписи из штампа времени на подпись.
LocalSigningTime	Время (UTC) подписи по локальным часам компьютера, на котором была создана подпись. Значение берётся из атрибута SigningTime.

1.5.4. VerifyParams

Перечисление содержит возможные значения дополнительных параметров проверки подписи.

Таблица 10. Поля перечисления VerifyParams

Имя	Описание
Hash	В качестве данных для проверки подписи передаётся хэш-значение от исходного документа.
SignatureId	Идентификатор подписи.
SignatureIndex	Порядковый номер подписи (начиная с 1).
VerifyAll	Проверить все подписи в документе.
HashAlgorithm	Идентификатор алгоритма хеширования. Допустимые значения: ➢ GOST R 34.11-94 ➢ GR 34.11-2012 256 ➢ GR 34.11-2012 512 ➢ SHA-1 ➢ SHA-256 ➢ SHA-384 ➢ SHA-512 ➢ MD5
VerifyPKCS7	Флаг, указывающий на то, что в параметре document передан PKCS#7
ExtractContent	Флаг, указывающий на то, что необходимо вернуть Content присоединенной CMS-подписи

1.6. Указание подписи для проверки

Функции VerifySignature и VerifyDetachedSignature позволяют указать подпись, которую требуется проверить. Указание подписи для проверки осуществляется через словарь дополнительных параметров **verifyParams**. Указать подпись для проверки можно по её идентификатору или порядковому номеру. Если в документе найдено более одной подписи, а словарь дополнительных параметров с указанием подписи не передан, то сервис вернёт ошибку.

1.6.1. Указание подписи формата MSOffice

Подпись формата MSOffice можно указать через её порядковый номер (SignatureIndex) или по идентификатору (SignatureId). В качестве значения SignatureId необходимо передать значение атрибута Id узла Signature. Порядковые номера и идентификаторы подписей можно получить, вызвав метод GetSignersInfo.

1.6.2. Указание подписи формата XMLDSig

Подпись формата XMLDSig можно указать через её порядковый номер (SignatureIndex) или по идентификатору (SignatureId). В качестве значения SignatureId необходимо передать значение атрибута Id узла Signature. Порядковые номера и идентификаторы подписей можно получить, вызвав метод GetSignersInfo.

1.6.3. Указание подписи формата PDF

Подпись формата PDF можно указать через её порядковый номер (SignatureIndex) или по имени (SignatureId). Порядковые номера и имена подписей можно получить, вызвав метод GetSignersInfo.

Пример проверки PDF подписи по имени:

```
SignersInfo results = srvClient.GetSignersInfo(SignatureType.PDF, signature);
SignerInfo si = results.SignerInfoList[0];
VerificationResult verifyResult =
                    srvClient.VerifySignature(
                              SignatureType.PDF
                              signature,
                              new Dictionary<VerifyParams, string>()
                                {
                                  { VerifyParams.SignatureId, si.Id }
                                });
```

1.6.4. Указание подписи формата CMS

Подпись формата CMS можно указать по её порядковому номеру (SignatureIndex). Порядковый номер подписи имеет формат: x_1,x_2, где x_1 – порядковый номер родительского узла подписи, x_2 – порядковый номер узла подписи, который требуется проверить. Порядковый номер подписи можно получить, вызвав метод GetSignersInfo. Если требуется проверить подпись первого уровня, то в значении порядкового номера подписи необходимо передать только порядковый номер подписи x_2.

Пример проверки CMS подписи по порядковому номеру:

```
SignersInfo results = srvClient.GetSignersInfo(SignatureType.CMS, signature);
string sigIndex = string.Empty;
SignerInfo si = results.SignerInfoList[0];
if (string.IsNullOrEmpty(si.ParentId))
{
     // Проверяем подпись первого уровня.
     sigIndex = si.Index.ToString();
}
else
{
     // Ищем в результатах разбора документа узел с идентификатором родительского
узла
     List<SignerInfo> parents = results.SignerInfoList.Where(x => x.Id ==
si.ParentId).ToList();
```

```
    // Формируем порядковый номер узла подписи второго уровня (заверяющей подписи)
    sigIndex = parents[0].Index + "," + si.Index.ToString();
}
VerificationResult verifyResult =
                    srvClient.VerifySignature(
                              SignatureType.CMS
                              signature,
                              new Dictionary<VerifyParams, string>()
                                {
                                  { VerifyParams.SignatureIndex, sigIndex }
                                });
```

2. Описание интерфейса REST для работы с сервисом проверки подписи «КриптоПро SVS»

REST-интерфейс предоставляет конечные точки для доступа к следующим функциям Сервиса Проверки Подписи:

- Проверка ЭП документа;
- Проверка сертификата;
- Получение информации о подписантах;
- Получение политики Сервиса Проверки Подписи.

В рамках REST API взаимодействие с Сервисом Проверки Подписи осуществляется посредством HTTP-запросов.

Общий префикс для всех конечных точек сервиса: **https://<hostname>/<ApplicationName>/rest**, где

- <hostname> – адрес сервера, на котором расположен экземпляр приложения Сервиса Проверки Подписи,
- <ApplicationName> – название веб-приложения Сервиса Проверки Подписи.

2.1. Формат входных и возвращаемых значений

Входные параметры могут передаваться в теле самого запроса. Все объекты передаются и возвращаются в формате JSON, списки возвращаются в виде массивов JSON объектов. Массивы байтов передаются в виде строк в кодировке BASE64.

2.2. Конечные точки сервиса проверки подписи

2.2.1. Конечная точка Certificates

Данная конечная точка позволяет получить доступ к функции, осуществляющей проверку действительности сертификата.

Таблица 11. Конечная точка Certificates

HTTP метод	**Путь**	**Описание**
POST	api/certificates	Проверка сертификата
Параметры	**Тип**	**Описание**
	Certificate	Объект, содержащий сертификат
Возвращаемое значение	**Тип**	**Описание**
	VerificationResultRest	Результат проверки сертификата

Пример выполнения метода:

Запрос

```
POST http://simdss.cryptopro.ru/verify/rest/api/certificates HTTP/1.1
Content-Type: application/json
Cache-Control: no-cache
Postman-Token: 2487016b-2e14-4763-85fa-c00a88214a18
      {
```

```
      "Content":
"MIIIgDCCCC+…49Ei5YNWEBytFwvzPOigd1rJjpHMwFGPVRv0maLh9dZXAiImx7tEm4="
      }
```

Ответ

```
      {
          "Message": null,
          "Result": true,
    "SignerCertificate": "MIIIgDCCCC+ …  ZXAiImx7tEm4=",
          "SignerCertificateInfo": {
              "SubjectName": "SN=Иванов, G=Иван Иванович, I=И.И., T=Инженер
технической поддержки, STREET=\"ул. Сущёвский вал, д. 18\", CN=Тестовый пользователь
simdss, OU=Отдел тестирования, O=\"ООО \"\"КРИПТО-ПРО\"\"\", L=Москва, S=77 Москва,
C=RU, E=ivanov@cp.ru, ИНН=334567890110, СНИЛС=33456789011, ОГРНИП=334567890110000,
ОГРН=3345678901111",
              "IssuerName": "CN=\"Тестовый УЦ ООО \"\"КРИПТО-ПРО\"\" (УЦ 2.0)\",
O=\"ООО \"\"КРИПТО-ПРО\"\"\", C=RU, L=Москва, E=info@cryptopro.ru, ИНН=007717107991,
ОГРН=1037700085444",
              "NotBefore": "2018-01-31T13:50:34",
              "NotAfter": "2018-04-30T14:00:34",
              "SerialNumber": "124D455D1500D780E8117506E333DDF4",
              "Thumbprint": "FD4598DBE0CAC5AA697998106C364A9ED8D2CB47"
          }
      }
```

2.2.2. Конечная точка Policy

Данная конечная точка позволяет получить доступ к политике (настройкам) Сервиса Проверки Подписи.

Таблица 12. Конечная точка Policy

HTTP метод	**Путь**	**Описание**
GET	api/policy	Получение политики Сервиса Проверки Подписи
Параметры	**Тип**	**Описание**
Параметры не требуются		
Возвращаемое значение	**Тип**	**Описание**
	VsPolicy	Политика Сервиса Проверки Подписи

Пример выполнения метода:

Запрос

```
GET http://simdss.cryptopro.ru/verify/rest/api/policy HTTP/1.1
Cache-Control: no-cache
Postman-Token: fd7121ce-c03c-4d12-b01c-29b9441828c7
```

Ответ

```
{
 "AllowedSignatureTypes": null,
 "SignatureDescriptions": [
```

```
    {
        "SignatureType": "PDF",
        "FileExtensions": [
            "pdf"
        ]
    },
    {
        "SignatureType": "MSOffice",
        "FileExtensions": [
            "docx"
        ]
    },
    {
        "SignatureType": "XMLDSig",
        "FileExtensions": [
            "xml"
        ]
    },
    {
        "SignatureType": "CMS",
        "FileExtensions": [
            "sig",
            "*"
        ]
    },
    {
        "SignatureType": "GOST3410",
        "FileExtensions": [
            "*"
        ]
    }
 ]
}
```

2.2.3. Конечная точка Signatures

Данная конечная точка позволяет получить доступ к функции, осуществляющей проверку ЭП документа.

Таблица 13. Проверка ЭП

HTTP метод	**Путь**	**Описание**
POST	api/signatures	Проверка ЭП документа или сертификата
Параметры	**Тип**	**Описание**
	SignedDocument	Документ, ЭП которого необходимо проверить
	Тип	**Описание**

Возвращаемое значение	VerificationResultRest	Результат проверки ЭП или сертификата

Пример выполнения метода:

Запрос

```
POST http://simdss.cryptopro.ru/verify/rest/api/signatures HTTP/1.1
Content-Type: application/json
Cache-Control: no-cache
Postman-Token: 547f407e-4ee1-43fa-8336-3ffdeb53151b

{
                "SignatureType": 2,
                "Content": "MIIFpAYJKoZIhvcNAQ … 4JXqpw="
}
```

Ответ

```
[
    {
        "Message": "Не удалось проверить подпись CAdES-BES. Ошибка: [Не удается построить цепочку сертификатов для доверенного корневого центра]. Код: [0x800b010a].Не удалось построить цепочку для сертификата, на ключе которого подписано сообщение.",
        "Result": false,
        "SignerCertificate": "MIIDEjCCAsGgAwIB … QCUHA==",
        "SignerCertificateInfo": {
            "SubjectName": "CN=Тестировщик",
            "IssuerName": "CN=CRYPTO-PRO Test Center 2, O=CRYPTO-PRO LLC, L=Moscow, C=RU, E=support@cryptopro.ru",
            "NotBefore": "2017-11-14T13:35:27",
            "NotAfter": "2018-02-14T13:45:27",
            "SerialNumber": "120022992D15DD6D649786F4C800000022992D",
            "Thumbprint": "BFDC90703E8CD7A242681BC32C22CCDA7DE15556"
        },
        "SignatureInfo": {
            "CAdESType": "BES",
            "LocalSigningTime": "2018-04-24T10:08:59"
        }
    }
]
```

2.2.4. Конечная точка SignersInfo

Таблица 14. Получение информации о подписанте

HTTP метод	**Путь**	**Описание**
POST	api/signatures/signersInfo	Получение информации о подписанте
Параметры	**Тип**	**Описание**
	SignedDocument	Документ, ЭП которого необходимо проверить

Возвращаемое значение	Тип	Описание
	SignersInfo	Информация о подписанте

Пример выполнения метода:

Запрос

```
POST http://simdss.cryptopro.ru/verify/rest/api/signatures/signersInfo HTTP/1.1
Content-Type: application/json
Cache-Control: no-cache
Postman-Token: d37881fe-f5c6-4dc7-9262-566b0ddaa3c9

{
                "SignatureType": 2,
                "Content": "MIIFpAYJKoZIhv … u34JXqpw="
}
```

Ответ

```
{
    "SignerInfoList": [
        {
            "Id": "0",
            "ParentId": "",
            "Index": 1,
            "SignerCertificateInfo": {
                "SubjectName": "CN=Тестировщик",
                "IssuerName": "CN=CRYPTO-PRO Test Center 2, O=CRYPTO-PRO LLC,
L=Moscow, C=RU, E=support@cryptopro.ru",
                "NotBefore": "2017-11-14T13:35:27",
                "NotAfter": "2018-02-14T13:45:27",
                "SerialNumber": "120022992D15DD6D649786F4C800000022992D",
                "Thumbprint": "BFDC90703E8CD7A242681BC32C22CCDA7DE15556"
            }
        }
    ],
    "AdditionalInfo": {
        "Content": null
    }
}
```

2.3. Типы данных

2.3.1. SignedDocument

Объект содержит документ, ЭП которого необходимо проверить.

Таблица 15. Описание свойств объекта SignedDocument

Свойство	Тип	Описание
SignatureType	SignatureType	Формат подписи SignatureType. Обязательный параметр.
Source	byte[]	Исходный документ. Используется только для проверки подписи формата GOST3410 и отделенной подписи форматов CMS и CAdES.
Certificate	byte[]	Сертификат подписанта. Используется только для проверки подписи формата GOST3410.
Content	byte[]	Подписанный документ или значение подписи для проверки. Обязательный параметр.
VerifyParams	Dictionary<VerifyParams, string>	Словарь дополнительных параметров проверки подписи. Необязательный параметр.
CertVerifiersPluginsIds	List<int>	Список идентификаторов плагинов для дополнительных проверок сертификата, используемого при подписи. Если CertVerifiersPluginsIds не передан в запросе или явно указан как NULL, то при проверке будут использоваться плагины, у которых CheckByDefaultRequired = true. Если CertVerifiersPluginsIds проинициализирован, то для проверки будут использованы плагины с указанными идентификаторами. Если список пустой — дополнительная проверка не выполняется.

2.3.2. Certificate

Объект содержит информацию о сертификате, статус которого необходимо проверить.

Таблица 16. Свойства объекта Certificate

Свойство	Тип	Описание
Content	byte[]	Сертификат.
CertVerifiersPluginsIds	List<int>	Список идентификаторов плагинов для дополнительных проверок сертификата. Если CertVerifiersPluginsIds не передан в запросе или явно указан как NULL, то при проверке будут использоваться плагины, у которых CheckByDefaultRequired = true. Если CertVerifiersPluginsIds проинициализирован, то для проверки будут использованы плагины с указанными идентификаторами. Если список пустой — дополнительная проверка не выполняется.

2.3.1. VerificationResultRest

Объект содержит информацию о результате проверки ЭП или сертификата.

Таблица 17. Свойства объекта VerificationResultRest

Свойство	Тип	Описание
Message	string	Суммарная информация о результатах проверки подписи.
Result	bool	Результат проверки подписи или сертификата.
SignerCertificate	byte[]	Сертификат.
SignerCertificateInfo	Dictionary<CertificateInfoParams, string>	Набор сведений о подписанте.
SignatureInfo	Dictionary<SignatureInfoParams	Дополнительные сведения о подписи.

2.3.1. VsPolicyRest

Объект содержит политику Сервиса Проверки Подписи.

Таблица 18. Свойства объекта VsPolicyRest

Свойство	Тип	Описание
SignatureDescriptions	IList<SignatureTypeDescription>	Форматы подписи, которые можно проверить.
CertificateVerifiers	List<CertificateVerifier>	Зарегистрированные плагины для дополнительных проверок сертификатов.

2.3.2. SignatureTypeDescription

Объект содержит описание поддерживаемых форматов подписи.

Таблица 19. Свойства объекта SignatureTypeDescription

Свойство	Тип	Описание
SignatureType	SignatureType	Перечисление, содержащее возможные форматы подписи.
FileExtensions	IList<string>	Связанные расширения файлов.

2.3.3. SignersInfo

Объект содержит сведения обо всех подписантах, а также дополнительную информацию о подписанном документе.

Таблица 20. Свойства объекта SignersInfo

Свойство	Тип	Описание
SignerInfoList	IList<SignerInfo>	Информация о найденных подписях.
AdditionalInfo	AdditionalSignedDocumentInfo	Дополнительная информация о подписанном документе.

2.3.4. SignerInfo

Описание свойств объекта представлено в **Таблица 3**.

2.3.5. AdditionalSignedDocumentInfo

Таблица 21. Свойства объекта AdditionalSignedDocumentInfo

Свойство	Тип	Описание
Content	byte[]	Содержимое присоединенной подписи в формате CMS.

2.3.6. CertificateVerifier

Таблица 22. Свойства объекта CertificateVerifier

Свойство	Тип	Описание
ID	int	Идентификатор плагина.
ClassName	string	Имя класса, который реализует интерфейс ISVSCertificateVerifier.
AssemblyName	string	Полный путь до файла со сборкой плагина. В качестве значения данного параметра можно указать полный путь до файла со сборкой, либо только имя dll-файла сборки, если плагин находится в следующей директории: <Путь установки>\Plugins\CertificatesVerifiers
PluginDescription	string	Описание плагина, которое отображается на Веб-интерфейсе Сервиса Проверки Подписи.
CheckByDefaultRequired	bool	Использовать ли по умолчанию плагин для проверки сертификата.
Parameters	Dictionary<string, string>	Дополнительные настройки плагина.

2.4. Перечислимые типы данных

2.4.1. SignatureType

Перечисление содержит поддерживаемые Сервером Проверки Подписи форматы подписи. Описание элементов приведено в **Таблица 7**.

2.4.2. CertificateInfoParams

Перечисление содержит возможные значения отображаемых данных о сертификате. Описание элементов приведено в **Таблица 8**.

2.4.3. SignatureInfoParams

Перечисление содержит возможные значения отображаемых данных о сертификате. Описание элементов приведено в **Таблица 9**.

2.4.4. VerifyParams

Перечисление содержит возможные значения дополнительных параметров проверки подписи. Описание элементов приведено в **Таблица 10**.

3. Создание и настройка модуля для дополнительной проверки сертификатов

Модуль для дополнительной проверки сертификатов должен представлять собой сборку .NET. Сборка должна содержать публичный класс, реализующий интерфейс **ISVSCertificateVerifier**. Интерфейс **ISVSCertificateVerifier** описан в сборке **CryptoPro.DSS.Common.dll**.

```
    public interface ISVSCertificateVerifier: IDSSPlugin
    {
        string PluginDefaultDescription { get; set; }

        int ID { get; set; }

        CertificateVerificationPluginResult Verify(byte[] certificate,
VerifyCertificateArgs args);
    }
```

Интерфейс **IDSSPlugin** предоставляет метод **Initialize**, который должен инициализировать подключаемый модуль по параметрам, передаваемым на вход данного метода в виде словаря.

```
public interface IDSSPlugin
    {
        void Initialize(IDictionary<string, string> parameters);
    }
```

Список дополнительных параметров задаётся при регистрации плагина с помощью командлета Add-VsCertificateVerifierPlugin. (см. ЖТЯИ.00094-01 90 09. КриптоПро SVS. Руководство Администратора).

Метод **Verify** принимает следующие параметры:

- **certificate** – сертификат для проверки в виде массива байт;
- **args** – объект, в который будут записаны дополнительные параметры.

При успешном завершении работы метод возвращает результат проверки сертификата.

Класс **VerifyCertificateArgs** описан в сборке **CryptoPro.DSS.Common.dll**.

```
    public class VerifyCertificateArgs
    {
        public IDictionary<string, string> Parameters;
    }
```

При завершении работы метода **Verify** необходимо заполнить возвращаемое значение **CertificateVerificationPluginResult**:

```
public class CertificateVerificationPluginResult
    {
        [DataMember]
        public bool bResult;

        [DataMember]
        public List<string> ErrorsList;
    }
```

- bResult — содержит результат проверки сертификата;
- ErrorsList — содержит сообщения об ошибках при проверке сертификата, если они есть.

4. Перечень таблиц