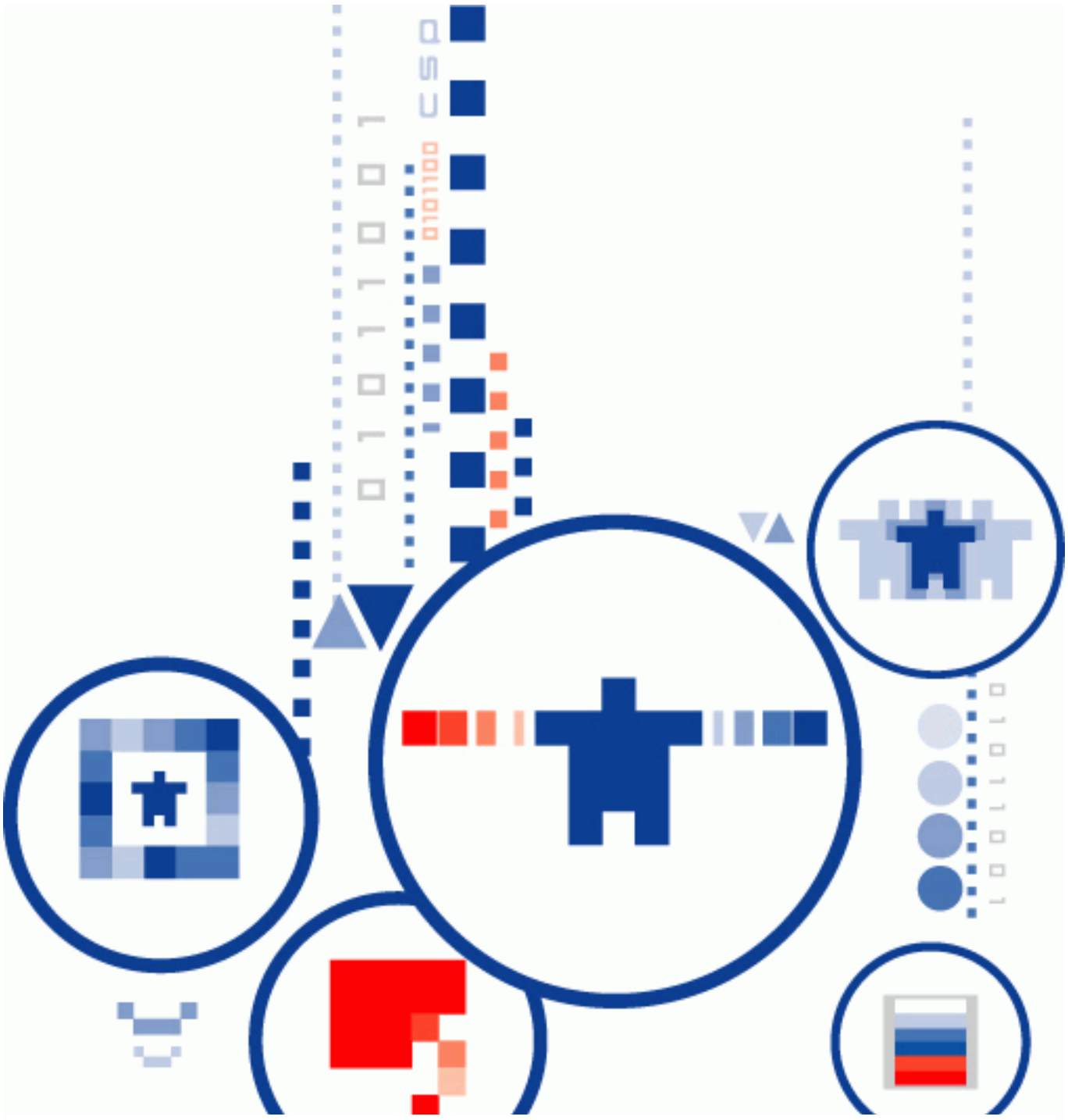
УТВЕРЖДЁН
ЖТЯИ.00094-01 20 01-ЛУ

ЖТЯИ.00094-01 90 10



Служба проверки сертификатов и электронной подписи

КриптоПро SVS

Версия 2.0

АННОТАЦИЯ

Настоящий документ содержит описание программного интерфейса сервиса проверки подписи «КриптоПро SVS», 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 2.0.1917.

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)

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 прикреплённую 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 прикреплённую 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 прикреплённую 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 прикреплённую 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 9.

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. SignersInfo

Класс описывает информацию о найденных в документе подписях. Описание полей класса приведено в Таблица 1.

Таблица 1. Описание полей класса SignersInfo

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

1.4.2. SignerInfo

Класс описывает информацию о подписи в документе. Описание полей класса приведено в Таблица 2.

Таблица 2. 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.

1.4.3. VsPolicy

Класс описывает настройки Сервиса Проверки Подписи. Описание полей класса приведено в Таблица 3.

Таблица 3. Описание полей класса VsPolicy

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

1.4.4. VerificationResult

Класс описывает результат проверки подписи или сертификата. Описание полей класса приведено в Таблица 4.

Таблица 4. 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 4.

Таблица 5. 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 6. Поля перечисления SignatureType

Имя	Описание
XMLDSig	Подпись документа в формате XMLDSig
GOST3410	Электронная подпись по ГОСТ Р 34.10 - 2001
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 7. 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 8. Поля перечисления SignatureInfoParams

Имя	Описание
CAdESType	Тип 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 9. 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

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. Перечень таблиц