ООО "Крипто-Про"

127 018, Москва, Улица Образцова, 38
Телефон: (095) 933 1168
Факс: (095) 289 4367
http://www.CryptoPro.ru
E-mail: info@CryptoPro.ru

ЭЦП процессор
Архитектура программного обеспечения

Содержание

Архитектура программного обеспечения

1. Вступление

1.1 Цель

Этот документ описывает архитектуру разрабатываемого программного продукта. Используя различные представления, он показывает различные аспекты системы и предназначен, чтобы собрать и преподнести все значимые для проекта архитектурные решения.

2. Описание документа

Данный документ описывает общую архитектуру разрабатываемого программного обеспечения в рамках проекта "ЭЦП-процессор". Архитектура представлена следующими представлениями:

1. Диаграммы вариантов использования;
2. Логическое представление;

3. Описание архитектуры

Разрабатываемое программное обеспечение (далее – ЭЦП-процессор) предназначено для внедрения сертифицированного средства СКЗИ КриптоПро CSP в систему защищенного электронного документооборота, а также для унификации доступа к криптографическим функциям СКЗИ КриптоПро CSP в рамках проекта «Система защиты информации ЭДО».

Универсальный программный интерфейс компонента обеспечит в прикладных программах, написанных на языках программирования, поддерживающих COM-объекты, формирование электронно-цифровой подписи и ее проверку с использованием сертификатов открытых ключей и криптографических процедур, реализованных в соответствии с:

- ГОСТ Р34.10-94, ГОСТ Р34.11-94, ГОСТ 28147-89 (КриптоПро CSP 1.x),
- ГОСТ Р34.10-2001, ГОСТ Р34.11-2001, ГОСТ 28147-2001 (КриптоПро CSP 2.x)
- других провайдеров, в рамках технологии CSP, установленных на локальном компьютере.

4. Назначение

Программное обеспечение предназначено для выполнения следующих функций:

1. Формирование ЭЦП документа.
2. Проверка ЭЦП документа.
3. Получение информации о сертификате.
4. Поиск сертификата для формирования ЭЦП и получения информации о персоне.

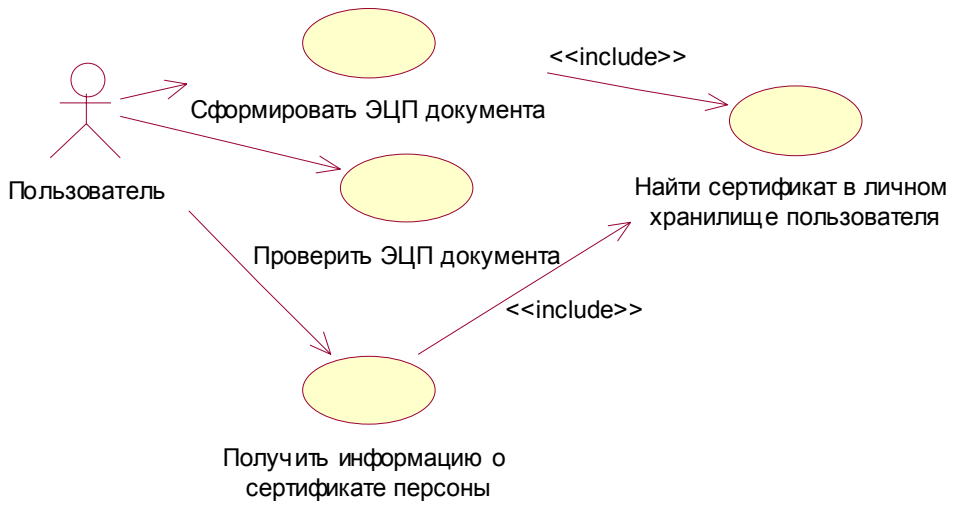
5. Диаграммы вариантов использования

5.1 Описание

Данный раздел описывает основные требования к программному обеспечению в виде стандартных диаграмм вариантов использования. Данные диаграммы достаточно полно описывают возможные варианты взаимодействия приложения ЭЦП-процессор (и его составляющих) с пользователем.

5.2 Общая диаграмма взаимодействия с пользователем

Ниже представлена общая диаграмма вариантов использования при взаимодействии пользователя с приложением ЭЦП-процессор.



В функциях формирования ЭЦП и получения информации о сертификате персоны используется общая функциональность для поиска сертификата в разделе «Личные» хранилища сертификатов текущего пользователя. Поэтому эта функциональная часть была оформлена отдельным вариантом использования.

Функциональные определения и требования вариантов использования представлены ниже.

5.2.1 Сформировать ЭЦП документа

Обеспечивает формирование ЭЦП на электронном документе. Сформированная ЭЦП не отделяется от подписанного электронного документа. Формирование ЭЦП осуществляется на закрытом ключе, соответствующем открытому ключу, содержащемуся в сертификате открытого ключа, найденному на компьютере пользователя в разделе «Личные» хранилища сертификатов текущего пользователя.

5.2.2 Проверить ЭЦП документа

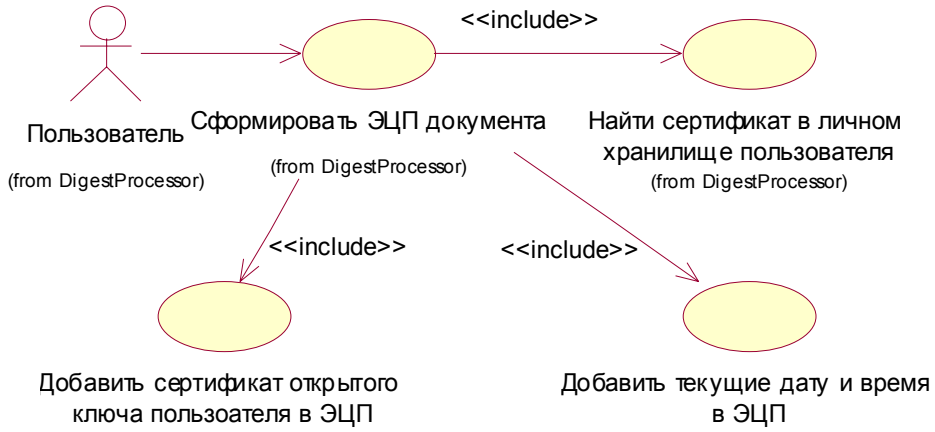
Осуществляет проверку корректности ЭЦП, содержащейся в электронном документе. Проверка осуществляется по передаваемому в ЭЦП сертификату открытого ключа.

5.2.3 Получить информацию о сертификате пользователя

Данная функция выполняет разбор сертификата открытого ключа и представление его в виде массивов данных. В качестве объекта разбора выступает или сертификат открытого ключа, переданного в качестве входных параметров или сертификат открытого ключа, найденного на компьютере пользователя в разделе «Личные» хранилища сертификатов текущего пользователя.

5.3 Диаграмма формирования ЭЦП документа.

Ниже приведена детализированная диаграмма формирования ЭЦП документа по запросу от пользователя.



Функциональные определения и требования вариантов использования представлены ниже.

5.3.1 Добавить сертификат открытого ключа пользователя в ЭЦП

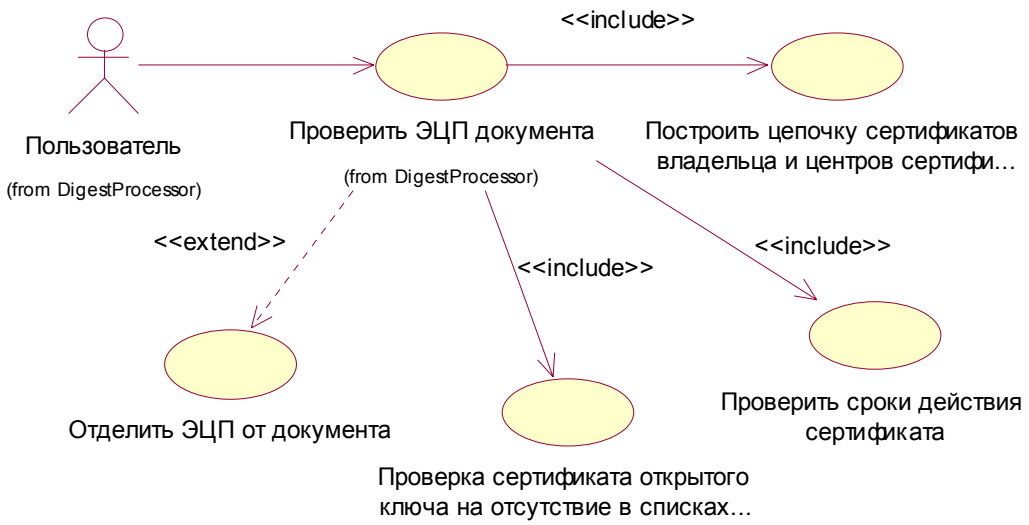
После формирования ЭЦП сертификат открытого ключа текущего пользователя должен добавляться (присоединяться) к ЭЦП.

5.3.2 Добавить текущие дату и время в ЭЦП

При формировании ЭЦП в подпись вносится атрибут, содержащий текущее системное время (дата и время) компьютера, на котором происходит формирование ЭЦП.

5.4 Диаграмма проверки ЭЦП документа

Ниже приведена детализированная диаграмма проверки ЭЦП документа по запросу от пользователя.



Функциональные определения вариантов использования представлены ниже.

5.4.1 Построить цепочку сертификатов владельца и центров сертификации

В рамках данной функции должна выполняться проверка валидности сертификатов открытых ключей, входящих в цепочку сертификатов ЭЦП (от сертификата пользователя до корневого центра сертификации).

5.4.2 Проверить сроки действия сертификата

Проверка сроков действия сертификатов открытого ключа из построенной цепочки сертификатов. Для проверки используется текущее системное время компьютера, на котором выполняется данная функция. В случае если текущее системное время компьютера находится в диапазоне срока действия сертификата, определенного в атрибутах сертификата, результат проверки считается положительным. В противном случае, результат проверки - отрицательный.

5.4.3 Проверка сертификата открытого ключа на отсутствие в списках отозванных сертификатов (CRL)

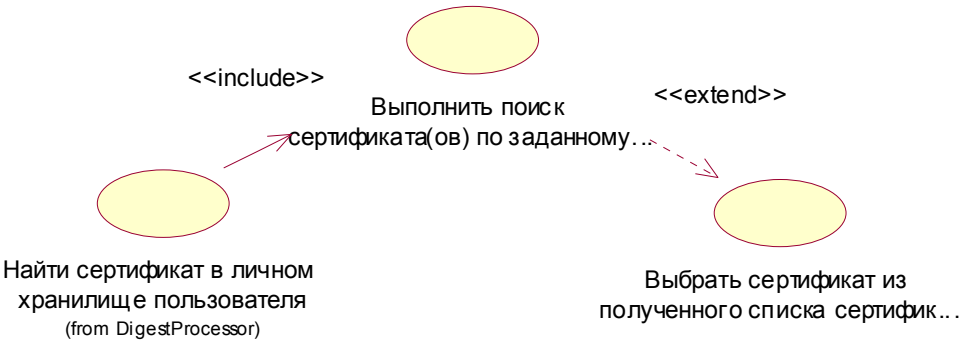
Проверка осуществляется по CRL, получаемых по адресу (URL) из точки распределения CRL (CDP), являющейся атрибутом сертификата открытого ключа, или из локально установленной версии CRL (выбор определяется параметров функции). При наличии проверяемого сертификата открытого ключа в списке отозванных сертификатов, результат проверки признается отрицательным. В противном случае результат данной проверки признается положительным. При недоступности CRL результат проверки считается отрицательным. При пустом значении CDP результат проверки считается положительным.

5.4.4 Отделить ЭЦП от документа

Данная функция может выполнять операцию «снятия» ЭЦП с документа, т.е. отделения ЭЦП из электронного документа, содержащего данную ЭЦП.

5.5 Диаграмма поиска сертификата

Ниже приведена детализированная диаграмма поиска сертификата в хранилище.



Поиск закрытого ключа для формирования ЭЦП осуществляется среди цифровых удостоверений (Digital ID), установленных на данном компьютере в двух хранилищах:

- Хранилище сертификатов «MY» текущего пользователя.
- Хранилище сертификатов «MY» текущего компьютера.

Функциональные определения вариантов использования представлены ниже.

5.5.1 Выполнить поиск сертифика(ов) по заданному критерию

Поиск сертификата для формирования ЭЦП осуществляется по следующим критериям:

- Для выбора хранилища используется параметр StoreLocation : 0 – поиск в хранилище сертификатов «MY» текущего пользователя, 1 - поиск в хранилище сертификатов «MY» текущего компьютера. По умолчанию значение параметра – 0.
- если передан серийный номер сертификата SN, то происходит однозначный поиск по серийному номеру;
- соответствие значения хотя бы одной пары OID+DN из Extended Key Usage сертификата открытого ключа и значениям OID и DN, переданных в качестве параметров.

5.5.2 Выбрать сертификат из полученного списка сертификатов.

При наличии более чем одного сертификата, удовлетворяющего приведенному выше критерию, выбор сертификата осуществляется пользователем посредством диалогового интерфейса, предоставляемого пользователю программным модулем. В диалоговом окне должны быть отображены следующие поля сертификатов:

- Subject.DN
- ExtendedKeyUsage.OID
- IssuerName
- ValidFromDate
- ValidToDate

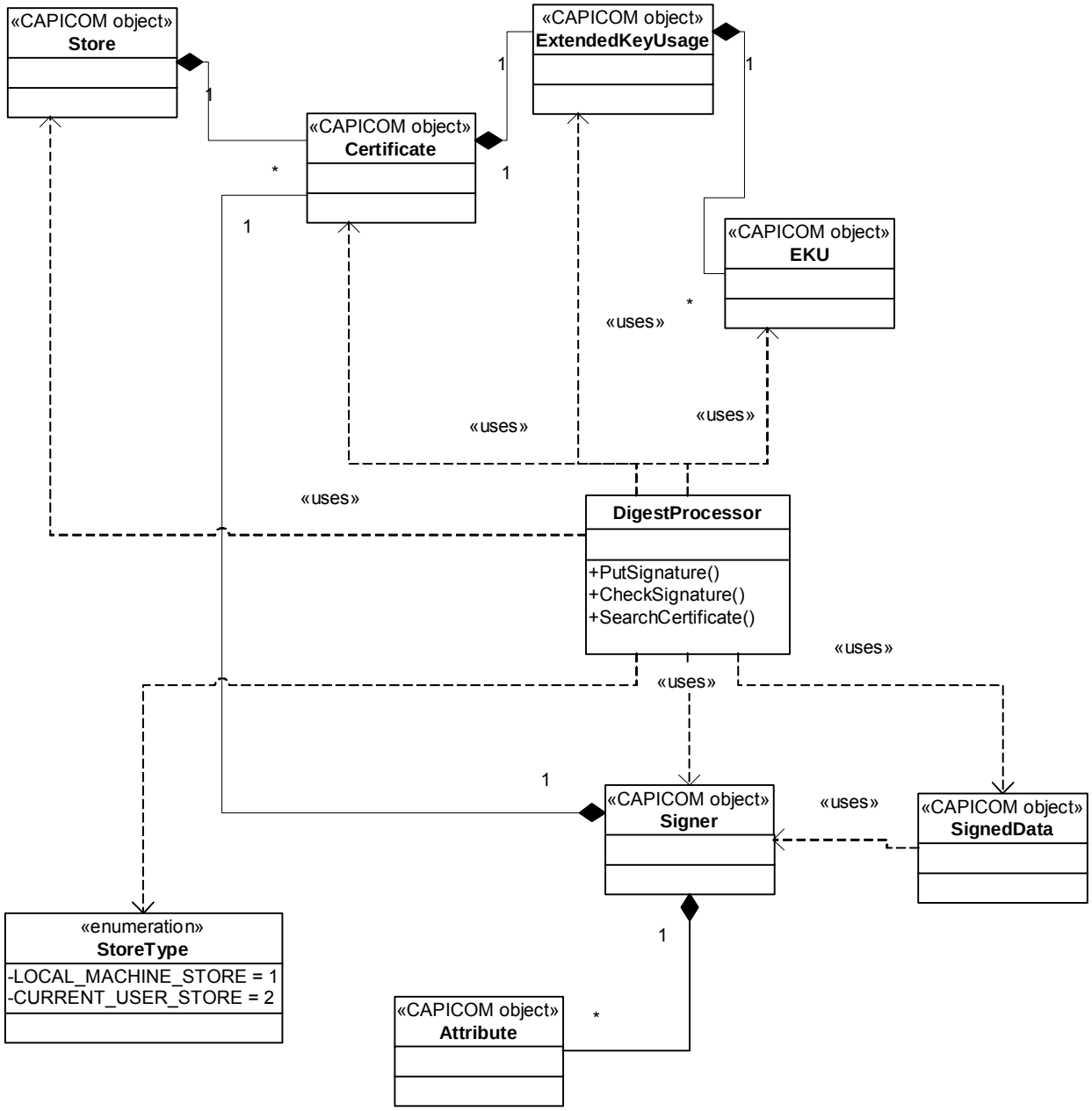
6. Логическое представление

6.1 Описание

Логическое представление приложения «ЭЦП-процессор» описано в виде диаграммы классов.

6.2 Компонент "Processor"

Данный компонент предназначен для формирования цифровой подписи, её проверки и снятия с документа, получения информации о сертификате.



6.2.1 Методы компонента

Метод	**Описание:**
PutSignature	Производит поиск сертификата удовлетворяющего указанным пользователем характеристикам и подписывает с его помощью документ.
CheckSignature	Производит проверку подписанного документа, при необходимости отделяет подписанный документ и сертификат. Определяет дату и время подписания.
SearchCertificate	Возвращает информацию о сертификате, найденном по указанным пользователем характеристикам, или переданном в виде строки.

6.2.1.1 Метод PutSignature

Производит поиск сертификата удовлетворяющего указанным пользователем характеристикам и подписывает с его помощью документ. Вносит в атрибут цифровой подписи дату и время подписания документа.

Параметры:

Название		**Тип**	**Значение по умолчанию**	**Описание**
Document	[in]	BSTR	-	Подписываемый документ.
SerialNumber	[in]	BSTR	“”	Серийный номер искомого сертификата
OID	[in]	BSTR	-	OID искомого сертификата.
DN	[in]	BSTR	“”	Строка DN (Distinguished Name) определяющая поле Subject искомого сертификата.
StorageType	[in]	StorageType	CURRENT_USER_STORE	Определяет расположение хранилища, сертификатов в котором будет производиться поиск.
FORMATUsage	[in]	FORMATUsage	FORMAT_BASE64	Определяет формат подписанного документа

Возвращаемые значения:

Название		**Тип**	**Описание**
SignedDocument	[out]	VARIANT (VT_BSTR)	Подписанный документ.
ErrorInfo	[out]	VARIANT (VT_BSTR)	Описание ошибки.
Result	[out, retval]	long	Код ошибки: 0 – нет ошибки, в противном случае код ошибки - HRESULT.

Примечания:

1. Тип StorageType является перечислением, имеющим следующие значения:

Название	Значение	Описание
CURRENT_USER_STORE	0	Хранилище сертификатов текущего пользователя.
LOCAL_MACHINE_STORE	1	Хранилище сертификатов локальной машины.

2. Тип FORMATUsage является перечислением, имеющим следующие значения:

Название	Значение	Описание
FORMAT_BASE64	0	Подписанный документ возвращается в формате Base-64.
FORMAT_BINARY	1	Подписанный документ возвращается в битовой последовательности.

6.2.1.2 Метод CheckSignature

Производит проверку подписанного документа, при необходимости отделяет подписанный документ и сертификат. Определяет дату и время подписания.

Параметры:

Название		Тип	Значение по умолчанию	Описание
SignedDocument	[in]	BSTR	-	Подписанный документ
RemoveSign	[in]	SignRemoval	DONT_REMOVE_SIGNATURE	Признак снятие цифровой подписи.
UseCRLFromAddress	[in]	CRLUsing	DONT_USE_CRL_FROM_CDP	Признак использования локального CRL.
SubjectName	[in]	BSTR	“”	Имя издателя сертификата.
StorageType	[in]	StorageType	CURRENT_USER_STORE	Определяет расположение хранилища, сертификатов в котором будет производиться поиск CRL.

Возвращаемые значения:

Название		Тип	Описание
DateAndTime	[out]	VARIANT (VT_BSTR)	Дата и время подписания.
Certificate	[out]	VARIANT (VT_BSTR)	Сертификат открытого ключа владельца ЭЦП.
Document	[out]	VARIANT (VT_BSTR)	Исходный документ.

ErrorInfo	[out]	VARIANT (VT_BSTR)	Описание ошибки.
Result	[out, retval]	long	Код ошибки: 0 – нет ошибки, в противном случае код ошибки - HRESULT.

Примечания:

1. Использование значений типа VARIANT связано с особенностью работы с несколькими возвращаемыми значениями в языках VBScript и JScript.

2. Тип SignRemoval является перечислением, имеющим следующие значения:

Название	**Значение**	**Описание**
DONT_REMOVE_SIGNATURE	0	Не снимать ЭЦП при проверке подписанного документа.
REMOVE_SIGNATURE	1	Снимать ЭЦП при проверке подписанного документа.

3. Тип CRLUsing является перечислением, имеющим следующие значения:

Название	**Значение**	**Описание**
DONT_USE_CRL_FROM_CDP	0	Использовать локальную копию CRL.
USE_CRL_FROM_CDP	1	Обновлять CRL, используя адрес из сертификата.

6.2.1.3 Метод SearchCertificate

Возвращает информацию о сертификате, найденном по указанным пользователем характеристикам, или переданном в виде строки.
Параметры:

Название		**Тип**	**Значение по умолчанию**	**Описание**
SearchCertificate	[in]	SearchType	DONT_PERFORM_SEARCH	Признак поиска.
Certificate	[in]	BSTR	“”	Сертификат, информацию о котором необходимо получить.
SerialNumber	[in]	BSTR	“”	Серийный номер искомого сертификата
OID	[in]	BSTR	-	OID искомого сертификата.
DN	[in]	BSTR	“”	Строка DN (Distinguished Name) определяющая поле Subject искомого сертификата.
StorageType	[in]	StorageType	CURRENT_USER_STORE	Определяет расположение хранилища, сертификатов в котором будет

				производиться поиск.

Возвращаемые значения:

Название		**Тип**	**Описание**
KeyArray	[out]	VARIANT	Массив названий полей.
ValArray	[out]	VARIANT	Массив значений полей.
FieldCount	[out]	VARIANT (VT_UI4)	Размерность массивов.
ErrorInfo	[out]	VARIANT	Описание ошибки.
Result	[out, retval]	long	Код ошибки: 0 – нет ошибки, в противном случае код ошибки - HRESULT.

Примечания:

1. Параметры KeyArray и ValArray имеют VARIANT тип VT_ARRAY | VT_VARIANT. Каждый элемент массива имеет VARIANT тип VT_BSTR. Нижняя граница индекса массива равна 0.
2. Тип SearchType является перечислением, имеющим следующие значения:

Название	**Значение**	**Описание**
DONT_PERFORM_SEARCH	0	Возвращать информацию о сертификате, переданном в параметре Certificate
PERFORM_SEARCH	1	Возвращать информацию о сертификате найденном используя параметры SerialNumber, OID, DN.

6.2.2 Логика использования методов компонента

6.2.2.1 Метод PutSignature

Данный метод обеспечивает простановку ЭЦП под электронным документом, используя MS Crypto API, посредством вызова программных компонентов MS CAPICOM 1.0.

Сформированная ЭЦП не отделяется от подписанного электронного документа. Формирование ЭЦП осуществляется на закрытом ключе, соответствующем открытому ключу, содержащемуся в сертификате открытого ключа, найденному на компьютере пользователя в разделе «Личные» хранилища сертификатов текущего пользователя и локального компьютера (в зависимости от переданных параметров). Логика работы функции поиска сертификата в хранилище описана в п. 6.3.2.4.

При формировании ЭЦП в подпись вносится атрибут, содержащий текущее системное время (дата и время) компьютера, на котором происходит формирование ЭЦП.

Поиск закрытого ключа для формирования ЭЦП осуществляется среди цифровых удостоверений (Digital ID), установленных на данном компьютере в двух хранилищах:

- Хранилище сертификатов «MY» текущего пользователя.
- Хранилище сертификатов «MY» текущего компьютера.

6.2.2.2 Метод CheckSignature

Данный метод осуществляет проверку корректности ЭЦП, содержащейся в электронном документе по передаваемому в ЭЦП сертификату открытого ключа. Метод осуществляет проверку математической корректности ЭЦП и проверку валидности переданного сертификата, так же осуществляется проверка SubjectName издателя сертификата на соответствие значению, которое передано в параметрах. Если значение не передано, то проверка не производится.

Также данный метод может выполнять операцию «снятия» ЭЦП с документа, т.е. отделения ЭЦП из электронного документа, содержащего данную ЭЦП.

Проверка валидности сертификатов открытых ключей, входящих в цепочку сертификатов ЭЦП, при работе данного метода осуществляется по следующим критериям:

- проверка сроков действия сертификатов открытого ключа из построенной цепочки сертификатов. Для проверки используется текущее системное время компьютера, на котором выполняется данный метод. В случае если текущее системное время компьютера находится в диапазоне срока действия сертификата, определенного в атрибутах сертификата, результат проверки считается положительным. В противном случае, результат проверки - отрицательный.
- проверка сертификатов открытого ключа на отсутствие в списках отозванных сертификатов (CRL). Проверка осуществляется по CRL, получаемых по адресу (URL) из точки распределения CRL (CDP), являющейся атрибутом сертификата открытого ключа, или из локально установленной версии CRL (выбор определяется параметрами вызова метода). При наличии проверяемого сертификата открытого ключа в списке отозванных сертификатов, результат проверки признается отрицательным. В противном случае результат данной проверки признается положительным. При недоступности CRL результат проверки считается отрицательным. При пустом значении CDP результат проверки считается положительным.

При проверке валидности сертификатов открытых ключей, строится цепочка сертификатов, состоящая из сертификата открытого ключа пользователя-владельца ЭЦП и сертификатов открытых ключей корневого и промежуточных Центров сертификации. Каждый из сертификатов подвергается проверке по указанным выше критериям. В случае если хотя бы один из сертификатов цепочки не удовлетворяет указанным выше критериям, результат работы функции признается отрицательным (код возврата = 1).

6.2.2.3 Метод SearchCertificate

Данный метод выполняет разбор сертификата открытого ключа и представление его в виде массивов данных.

В качестве объекта разбора выступает или сертификат открытого ключа, переданного в качестве входных параметров или сертификат открытого ключа, найденного на компьютере пользователя в разделе «Личные» хранилища сертификатов текущего пользователя. Требования к функции поиска сертификата в хранилище описаны в п. 6.3.2.4.

6.2.2.4 Логика работы функции поиска сертификата

Поиск сертификата для формирования ЭЦП осуществляется по следующим

критериям:

- Для выбора хранилища сертификатов используется соответствующий параметр (см. п. 6.2.1.).
- Если передан серийный номер сертификата SN, то происходит однозначный поиск по серийному номеру;
- Если передан OID и DN, то соответствие значения хотя бы одной пары OID+DN из Extended Key Usage сертификатов открытого ключа (установленных на локальном компьютере) значениям OID и DN, переданных в качестве параметров;
- при наличии более чем одного сертификата (при поиске по OID и DN), удовлетворяющего приведенному выше критерию, выбор сертификата осуществляется пользователем посредством диалогового интерфейса, предоставляемого пользователю программным модулем. В диалоговом окне должны быть отображены следующие поля сертификатов:
    - Subject.DN
    - ExtendedKeyUsage.OID
    - IssuerName
    - ValidFromDate
    - ValidToDate

6.2.3 Правила создания и уничтожения компонента.

Правила создания компонент полностью соответствуют правилам создания COM компонентов.

Компонент может быть создан как с использованием ProgID – для скриптовых языков, так и с использование CLSID – для C++, Visual Basic и других языков поддерживающих подобный способ создания компонента.

Значение ProgID компонента: «DigestProcessor.Processor»
Значение CLSID компонента: AEEBAEF9-8A04-4777-B783-A17C21FCE50D

Уничтожение компонента производится также в соответствии со стандартной процедурой уничтожения COM компонентов.

Пример создания и уничтожения компонента средствами языка VBScript:

```
‘Объявление переменной объекта
Dim objDigestProc

‘Создание объекта по его ProgID
Set objDigestProc = CreateObject(“DigestProcessor.Processor”)

‘…

‘Уничтожение компонента
Set objDigestProc = Nothing
```

6.2.4 Ограничения использования компонента

Программный модуль «ЭЦП-Процессор» предназначен для функционирования в среде операционных систем MS Windows 95/98/Me/NT 4.0 с установленным MS Internet Explorer 5.0 и выше, а также MS Windows 2000 Prof/Server и MS Windows XP.

Кроме того, необходимо наличие в операционной системе следующих установленных программных компонентов:

- MS CAPICOM версии 1.0. или выше.
- Компонент для обновления списка отозванных сертификатов (CRL) на локальном компьютере с удостоверяющего центра (cpcrlupdate.dll).

7. Описание использования примеров компонента "ЭЦП Процессор"

Для начала работы с примером загрузите страницу DigestProcessor.html и выберете пункт меню "Пример". Далее следуйте приведённым ниже инструкциям.

7.1.1 Создание цифровой подписи

Стартовая страница примера предназначена для создания цифровой подписи на документе и указание необходимых для этого атрибутов. Для создания цифровой подписи используется сертификат, удовлетворяющий указанным пользователем параметрам. На странице расположены следующие поля:

1. "Подписываемый документ" – данное поле предназначено для ввода подписываемого текста. Подпись пустого документа не допустима о чём, по необходимости, сообщает соответствующее диалоговое окно.
2. "Серийный номер сертификата" – данное поле определяет серийный номер сертификата. Если серийный номер указан, то происходит однозначный поиск по его значению (т.е. значения параметров OID, "Ф.И.О.", "Должность", "Компания" и "Отдел" игнорируются).

3. "OID" - данное поле предназначено для выбора значения параметра сертификата "Object Identifier". Данное поле является обязательным для ввода, о чём, по необходимости, сообщает соответствующее диалоговое окно.

4. "Ф.И.О." – определяет значения соответствующего атрибута сертификата.

5. "Должность" – определяет значения соответствующего атрибута сертификата.

6. "Компания" – определяет значения соответствующего атрибута сертификата.

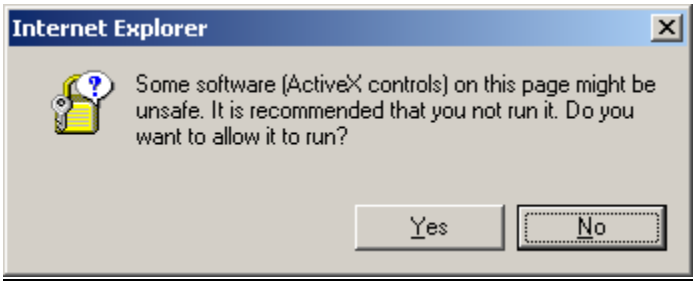
7. "Отдел" – определяет значения соответствующего атрибута сертификата.

8. "Хранилище сертификата" – определяет, в каком именно хранилище следует искать сертификат.

   Поля 3. - 6. является опциональными – в том случае если часть из них не заполнена, то их значение при поиске игнорируются.

   Заполните поля формы и нажмите кнопку "Подписать".

   После чего возникнет следующее диалоговое окно:



Для продолжения работы следует выбрать “Yes”. После чего будет произведена процедура подписи документа. При её успешном окончании будет открыта следующая страница примера.

7.1.2 Подписанный документ

Данная страница предназначена для отображения подписанного документа в base64 кодировке и проверки ЭЦП сделанной на предыдущем шаге.

Для осуществления проверки нажмите кнопку “Проверить подпись”

Условие проверки может быть усилено, для этого следует указать имя владельца сертификата, издателя сертификата ЭЦП. Для этого предназначено текстовое поле “Имя владельца сертификата, издавшего сертификат ЭЦП”.

Кроме того, может быть указано какой именно список отозванных сертификатов нужно применять при проверке сертификата ЭЦП. Выбор производится с помощью поля списком “Использование при проверке списка отозванных сертификатов”

После её нажатия кнопки “ Проверить подпись” также возникнет описываемое выше диалоговое окно. В данном случае это также следует выбрать “Yes”. В случае успешности проверки ЭЦП будет открыта следующая страница примера, в противном случае будет показан диалог с информацией об ошибке.

7.1.3 Проверка и снятие цифровой подписи

Эта страница отображает данные получения в результате снятие ЭЦП с документа, а именно:

1. Сам документ.
2. Сертификат в base64 кодировке.
3. Дата и время подписания.

Для получения подробной информации о сертификате, использованном для формирования ЭЦП, нажмите кнопку “Информация о сертификате”. После её нажатия также возникнет описываемое выше диалоговое окно. При этом также следует выбрать “Yes”.

7.1.4 Свойства сертификата

На этой странице вы можете увидеть свойства сертификата использованного для формирования ЭЦП на документе.

7.1.5 Приложение

Пример состоит из следующих файлов и папок:

	Описание
DigestProcessor.html	Страница описания компонента.
Examples\start.htm	Стартовая страница примера: “1. Создание цифровой подписи”.
Examples\ common.vbs	Файл, содержащий функции общего назначения, используемые в примере: создание и открытие страницы по шаблону; уведомление об ошибках.
Examples\Output	В данную папку помещаются страницы создаваемые в процессе работы примера.
Examples\Templates\details_templ.htm	Шаблон страницы примера “3. Проверка и снятие цифровой подписи”.
Examples\Templates\prop_templ.htm	Шаблон страницы примера “4. Свойства сертификата”.
Examples\Templates\verify_templ.htm	Шаблон страницы примера “2. Подписанный документ”.

8. Установка дистрибутива на локальный компьютер

8.1 Инструкция по настройке дистрибутивного пакета (ДП) для инсталляции «ЭЦП-Процессор» с веб-сервера

В данной инструкции рассмотрена настройка ДП на примере веб-сервера http://www.example.com
Структура каталогов примерного сайта такова:

URL	**File**	**Описание**
http://www.example.com/Products/DigestProcessor/	Setup.exe	Setup.exe – загрузчик.
http://www.example.com/Products/DigestProcessor/	DigestProcessor.msi	Дистрибутив
http://www.example.com/Products/InstMsi/Ansi	Instmsi.exe	Установщик ANSI Windows Installer 2.0.
http://www.example.com/Products/InstMsi/Unicode	Instmsi.exe	Установщик Unicode Windows Installer 2.0.

8.2 Пошаговая инструкция

1. Для настройки дистрибутива и загрузчика setup.exe в каталоге %InstallDir%\DigestProcessor\WebInstaller нужно отредактировать пакетный файл tun.bat: указать имя сервера и нужную структуру каталогов. Затем запустить его.

2. Запустить утилиту SignCode.Exe для подписи цифровым сертификатом файлов

Setup.exe и DigestProcessor.msi.

3. Отредактировать ссылку "DigestProcessor Installation" в файле http://www.example.com/Products/ DigestProcessor/DigestProcessor.html .

Пользователи, посетив страницу http://www.example.com/Products/DigestProcessor/DigestProcessor.html и нажав на ссылку "DigestProcessor Installation", могут запустить установку «ЭЦП-Процессор» либо выкачать на локальный диск файл setup.exe. При запуске setup.exe с локального диска пользователя будет предложено установить «ЭЦП-Процессор» с сервера.

8.3 Структура размещения Программных компонентов «ЭЦП-Процессор» на диске после локальной установки.

После установки пакета на стороне клиента, на диске должны быть созданы следующие директории:

(%InstallDir% - корневая директория установки)

%InstallDir%\DigestProcessor – Корневая директория «ЭЦП-Процессор».
%InstallDir%\DigestProcessor\Bin – Директория для бинарных файлов «ЭЦП-Процессор».
%InstallDir%\DigestProcessor\Doc – Директория для документации.
%InstallDir%\DigestProcessor\Examples – Директория для примеров.
%InstallDir%\DigestProcessor\WebInstaller – Директория содержит дистрибутивный пакет, утилиту для подписи цифровым сертификатом дистрибутивного пакета, утилиту настройки дистрибутивного пакета для инсталляции «ЭЦП-Процессор» с веб-сервера. Также в этой директории находиться документация по использованию этих утилит и настройки веб-сервера для инсталляции «ЭЦП-Процессор».

При установке дистрибутива на стороне сервера корневая директория «ЭЦП-Процессор» должна быть %InstallDir%\DigestProcessorWeb.

9. Перечень ошибок, возвращаемых при работе ЭЦП-процессор

9.1 Ошибки, возвращаемые компонентом CAPICOM

Обозначение/код ошибки	Описание
CAPICOM_E_ENCODE_INVALID_TYPE 0x80880100	An invalid encoding type was used. The valid encoding types are CAPICOM_ENCODE_BASE64 or CAPICOM_ENCODE_BINARY.
CAPICOM_E_EKU_INVALID_OID 0x80880200	The **OID** property of the **EKU** object cannot be set because the **Name** property is not set to CAPICOM_EKU_OTHER. To correct, set the **Name** property to CAPICOM_EKU_OTHER before setting the **OID** property.
CAPICOM_E_EKU_OID_NOT_INITIALIZED 0x80880201	The OID value of the **EKU** object has not been initialized.

Обозначение/код ошибки	Описание
	To initialize it, either set the **Name** property to anything other than CAPICOM_EKU_OTHER, or set the **Name** property to CAPICOM_EKU_OTHER and the **OID** property to a value.
CAPICOM_E_CERTIFICATE_NOT_INITIALIZED 0x80880210	The **Certificate** object has not been initialized. Usually, this happens when a **Certificate** object is instantiated, but somehow the object is not associated to a digital certificate. To associate the object to a digital certificate, either assign it to an existing **Certificate** object or call the **Import** method.
CAPICOM_E_CERTIFICATE_NO_PRIVATE_KEY 0x80880211	The **Certificate** object does not have an associated private key. This error code is returned when an attempt is made to sign data using the signer's private key. This usually means the **Certificate** object associated with the **Signer** object cannot be used for the signing operation.
CAPICOM_E_CHAIN_NOT_BUILT 0x80880220	The Cha**i**n object has not been initialized. To initialize, call the **Build** method.
CAPICOM_E_STORE_NOT_OPENED 0x80880230	The **Store** object has not been initialized. To initialize, call the **Open** method.
CAPICOM_E_STORE_EMPTY 0x80880231	The **Store** object does not contain any **Certificate** objects.
CAPICOM_E_STORE_INVALID_SAVE_AS_TYPE 0x80880232	Invalid *SaveAs* value for saving the store. Valid *SaveAs* values are CAPICOM_STORE_SAVE_AS_SERIALIZED or CAPICOM_STORE_SAVE_AS_PKCS7.
CAPICOM_E_ATTRIBUTE_NAME_NOT_INITIALIZED 0x80880240	The **Name** property of the **Attribute** object has not been initialized. To initialize, set the **Name** property.
CAPICOM_E_ATTRIBUTE_VALUE_NOT_INITIALIZED 0x80880241	The **Value** property of the **Attribute** object has not been initialized. To initialize, set the **Value** property.
CAPICOM_E_ATTRIBUTE_INVALID_NAME 0x80880242	The **Name** property of the **Attribute** object is invalid. The valid attribute names are: CAPICOM_AUTHENTICATED_ATTRIBUTE_SIGNING_TIME CAPICOM_AUTHENTICATED_ATTRIBUTE_DOCUMENT_NAME CAPICOM_AUTHENTICATED_ATTRIBUTE_DOCUMENT_DESCRIPTION
CAPICOM_E_ATTRIBUTE_INVALID_VALUE 0x80880243	The **Value** property of the **Attribute** object is invalid because the data type does not match what is intended to be stored, as indicated by the **Name** property.

Обозначение/код ошибки	Описание
	For example, if the **Name** property is set to CAPICOM_AUTHENTICATED_ATTRIBUTE_SIGNING_TIME, the data type must be a **DATE** type.
CAPICOM_E_SIGNER_NOT_INITIALIZED 0x80880250	The **Signer** object has not been initialized. To initialize, set the **Certificate** property.
CAPICOM_E_SIGNER_NOT_FOUND 0x80880251	The signer cannot be found in the **SignedData** object. This usually does not happen with **SignedData** object that was created by CAPICOM; however, if the **SignedData** object was created by a third-party product, the signer's certificate may not be included in the PKCS #7 structure.
CAPICOM_E_SIGN_NOT_INITIALIZED 0x80880260	The **SignedData** object has not been initialized. To initialize, set the **Content** property or call the **Verify** method.
CAPICOM_E_SIGN_INVALID_TYPE 0x80880261	The **SignedData** object contains an invalid type. This usually happen when trying to verify an enveloped message with a **SignedData** object or vice versa.
CAPICOM_E_SIGN_NOT_SIGNED 0x80880262	The **SignedData** object has not been signed. To sign, call the **Sign** method.
CAPICOM_E_INVALID_ALGORITHM 0x80880270	Invalid algorithm value for the **Name** property of the **Algorithm** object. Valid algorithm values for the **Name** property are: CAPICOM_ENCRYPTION_ALGORITHM_RC2 CAPICOM_ENCRYPTION_ALGORITHM_RC4 CAPICOM_ENCRYPTION_ALGORITHM_DES CAPICOM_ENCRYPTION_ALGORITHM_3DES
CAPICOM_E_INVALID_KEY_LENGTH 0x80880271	Invalid key length value for the **KeyLength** property of the **Algorithm** object. Valid key length values for the **KeyLength** property are: CAPICOM_ENCRYPTION_KEY_LENGTH_MAXIMUM CAPICOM_ENCRYPTION_KEY_LENGTH_40_BITS CAPICOM_ENCRYPTION_KEY_LENGTH_56_BITS CAPICOM_ENCRYPTION_KEY_LENGTH_128_BITS
CAPICOM_E_ENVELOP_NOT_INITIALIZED	The **EnvelopedData** object has not been

Обозначение/код ошибки	Описание
0x80880280	initialized. To initialize, set the **Content** property or call the **Decrypt** method.
CAPICOM_E_ENVELOP_INVALID_TYPE 0x80880281	The **EnvelopedData** object contains an invalid type. This usually happen when trying to verify a signed message with an **EnvelopedData** object or vice versa.
CAPICOM_E_ENVELOP_NO_RECIPIENT 0x80880283	There is no recipient specified in the **EnvelopedData** object when the **Encrypt** method of an **EnvelopedData** object is called. To add a recipient, call the **Recipients.Add** method.
CAPICOM_E_ENVELOP_RECIPIENT_NOT_FOUND 0x80880284	The recipient cannot be found in the **EnvelopedData** object. This usually does not happen with **EnvelopedData** object that was created by CAPICOM; however, if the **EnvelopedData** object was created by a third-party product, the recipient's certificate may not be included in the PKCS #7 structure.
CAPICOM_E_ENCRYPT_NOT_INITIALIZED 0x80880290	The **EncryptedData** object has not been initialized. To initialize, set the **Content** property or call the **Decrypt** method.
CAPICOM_E_ENCRYPT_INVALID_TYPE 0x80880291	The **EncryptedData** object is not a valid type. This usually means the data is corrupted.
CAPICOM_E_ENCRYPT_NO_SECRET 0x80880292	The secret of an **EncryptedData** object has not been initialized. To initialize, call the **SetSecret** method.
CAPICOM_E_NOT_SUPPORTED 0x80880900	The requested operation is not supported in the current platform.
CAPICOM_E_UI_DISABLED 0x80880901	When signing, the **Certificate** property of the **Signer** object has not been set, but the prompt for the user certificate has been disabled. Either enable the prompt by setting the **EnablePromptForCertificateUI** property of the **Settings** object, or set the **Certificate** property of the **Signer** object.
CAPICOM_E_CANCELLED 0x80880902	The operation has been canceled by the user. This happens when the user is prompted for permission to carry out certain operation, such as accessing the private key, and the user cancels the operation.
CAPICOM_E_INTERNAL	An internal error has occurred.

Обозначение/код ошибки	Описание
0x80880911	Contact Microsoft Technical Support for assistance.
CAPICOM_E_UNKNOWN 0x80880999	An unknown error has occurred. Collect as much information as possible and contact your vendor.

9.2 Ошибки, возвращаемые компонентом DigestProcessor

Обозначение/код ошибки	Возвращаемая строка с описанием ошибки	Описание
E_FAIL 0x80004005	Отсутствует сертификат, удовлетворяющий указанным условиям.	Возвращается при отсутствии сертификата с указанными параметрами поиска.
E_FAIL 0x80004005	SubjectName сертификата подписи не совпадает с переданным значением.	Возвращается при несовпадении SubjectName сертификата с переданным значением.
Код ошибки возвращенный CPCRLUpdate.dll	Ошибка обновления CRL: <<ОПИСАНИЕ ОШИБКИ>>	Возникает при ошибке обновления CRL
CAPICOM_E_CHAIN_NOT_BUILT 0x80880220	The certificate chain has not been built.	Возникает при ошибке построения цепочки сертификатов. См. ниже список причин ошибки, п. 9.3.
0x8009310B	ЭЦП в документе не найдена (ASN1 bad tag value met).	Возникает при нарушении целостности ЭЦП.
0x80093004	ЭЦП в документе не найдена (ASN1 bad tag value met).	Возникает при нарушении целостности ЭЦП.
	The parameter is incorrect.	Ошибка в передаваемых параметрах
0x80090006	Неправильная подпись	
0x80091007	Неправильное значение хеша	
0x80091004	Недопустимый тип криптографического сообщения	

9.3 Описание возможных причин возникновения ошибок при построении цепочки сертификатов

Описание причины

The current date is not within a certificate's valid period.
The time validity of a certificate in the chain falls outside the time validity of one or more of its verifying certificates.
One or more of the certificates in the chain has been revoked.
One or more of the certificates in the chain does not have a valid signature.
One or more of the certificates in the chain is not valid for its usage.
The *root certificate* of the chain is not trusted.
The revocation status of one or more of the certificates in the chain cannot be determined.
A certificate in the chain is used to certify a certificate that was used in its own certification.
The truest chain cannot be completed to a certificate in the Root store.
The chain depends upon a CTL that is not time-valid.
The chain depends upon a CTL that does not have a valid signature.
The chain depends upon a CTL that is not valid for its usage in the chain.