ViPNet Terminal 4. API ЭП в браузере Firefox

Руководство разработчика

infotecs

ФРКЕ.00154-01 33 01

Версия продукта 4.1.8

ОАО «ИнфоТеКС»

127287, г. Москва, Старый Петровско-Разумовский проезд, дом 1/23, строение 1

Тел: (495) 737-61-96 (горячая линия), 737-61-92, факс 737-72-78

Сайт компании «ИнфоТеКС»: http://www.infotecs.ru

Электронный адрес службы поддержки: hotline@infotecs.ru

Содержание

API электронной подписи в браузере Firefox **4**

Общие сведения 4

Требования к окружению 4

Функция подписи 5

Функция получения списка сертификатов 5

Пример использования функций 6

Формирование и проверка подписи с использованием тестовой веб-страницы **7**

Порядок формирования и проверки подписи 7

Требования к дополнительному ПО 7

Получение сертификата и закрытого ключа 8

Формирование подписи 9

Проверка подписи 10

Тестовая веб-страница 11

API электронной подписи в браузере Firefox

Общие сведения

Функция электронной подписи и сопряженные функции реализованы в плагине, который предоставляет API на уровень JavaScript, доступный для разработчиков веб-порталов.

Ниже представлена архитектура решения для разработчиков веб-порталов.

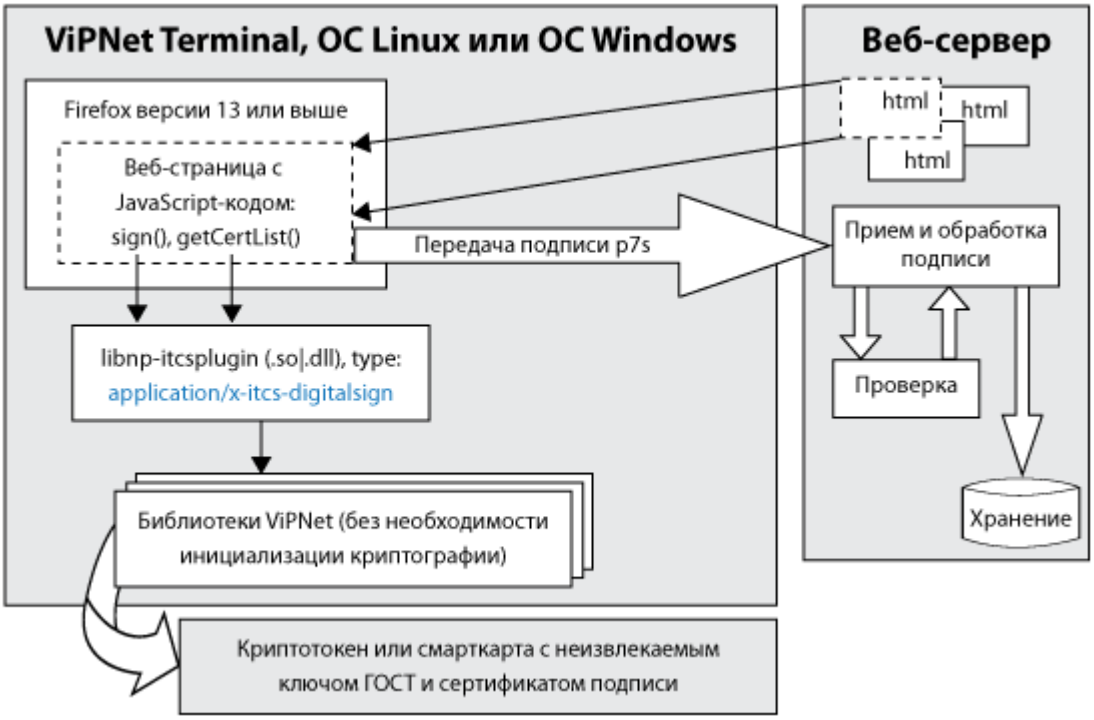

Рисунок 1. Архитектура решения для разработчиков веб-порталов

Требования к окружению

На клиентском рабочем месте должен быть установлен ViPNet Terminal или персональный компьютер с ОС Linux/Windows и набором библиотек ViPNet.

В качестве веб-браузера на рабочем месте должен быть установлен Firefox версии не ниже 13.

Функция подписи

Функция возвращает закодированную в base64 подпись в формате PKCS#7 данных на заданном сертификате, имеющемся на криптотокене.

- Вызов:

  ```
  Result = sign(Pin, Cert, Data, SignType)
  ```

- Параметры:
  - `Pin` — строка с ПИН-кодом единственного подключенного к компьютеру криптотокена;
  - `Cert` — сертификат в виде строки base64, закрытым ключом которого должны быть подписаны данные;
  - `Data` — строка с данными, подлежащими подписи (для бинарных данных это может быть строка base64);
  - `SignType` — строка с типом получаемой подписи:
    - `"p7s_attached"` — данные с подписью в формате PKCS#7;
    - `"p7s_detached"` — подпись в формате PKCS#7 (без данных).
- Возвращаемое значение:

  Строка с подписью заданного типа (для бинарных типов, например, p7s, в base64) или строка с сообщением об ошибке. Сообщение об ошибке начинается со строки «Error: », после которой следует детальная информация о произошедшей ошибке.

Функция получения списка сертификатов

Функция возвращает список сертификатов в формате X.509 base64, найденных на единственном подключенном к компьютеру криптотокене.

- Вызов:

  ```
  CertList = getCertList()
  ```

- Параметры: нет.
- Возвращаемое значение:

  Список строк, каждая из которых содержит закодированный в base64 сертификат X.509. Для извлечения информации о сертификатах из элементов списка можно использовать библиотеку http://kjur.github.com/jsrsasign.

Пример использования функций

Фрагмент HTML-кода для включения плагина:

```
<embed id="signer" type="application/x-itcs-digitalsign"
       hidden="true">
```

Фрагмент JavaScript-кода для электронной подписи данных:

```
var o = document.getElementById("signer");
if(o) {
    certList = o.getCertList();
    if(certList.length == 1) {resultingSignature = o.sign(pinCode, certList[0],
dataToSign, "p7s_detached");
      document.getElementById("signature").value = resultingSignature;
    }
    // else
    //   say error
}
```

Формирование и проверка подписи с использованием тестовой веб-страницы

Порядок формирования и проверки подписи

Тестовая веб-страница (на стр. 11) поможет вам отладить и доработать собственную веб-страницу, чтобы встроить электронную подпись в ваше программное обеспечение. Чтобы проверить работу плагина подписи, выполните следующие действия:

1. Подготовьте для использования криптотокен и установите на какой-либо компьютер программы ViPNet CSP и SoapUI (см. «Требования к дополнительному ПО» на стр. 7).
2. Получите сертификат и закрытый ключ, затем в программе ViPNet CSP проверьте наличие на криптотокене контейнера ключей и его содержимое (см. «Получение сертификата и закрытого ключа» на стр. 8).
3. На сетевом узле ViPNet Terminal с помощью тестовой веб-страницы сформируйте подпись и скопируйте ее на USB-носитель для дальнейшей проверки (см. «Формирование подписи» на стр. 9).
4. В программе SoapUI проверьте сформированную подпись (см. «Проверка подписи» на стр. 10).

Требования к дополнительному ПО

Для формирования и проверки подписи с использованием тестовой веб-страницы вам понадобится криптотокен с поддержкой алгоритма шифрования ГОСТ и следующее программное обеспечение:

- Драйвер для криптотокена.

  Перед использованием криптотокена необходимо установить соответствующий драйвер и выполнить инициализацию криптотокена. Для инициализации и последующей работы с криптотокеном потребуется ПИН-код. Для криптотокена ruToken-ЭЦП стандартный ПИН-код 12345678, для eToken-ГОСТ — 1234567890.

- Программа ViPNet CSP.

  Установочный файл программы ViPNet CSP вы можете бесплатно загрузить на сайте компании «ИнфоТеКС» (https://www.infotecs.ru/downloads/product_full.php?id_product=2096).

При настройке параметров установки убедитесь, что выбран компонент **Поддержка работы ViPNet CSP через MS Crypto API**.

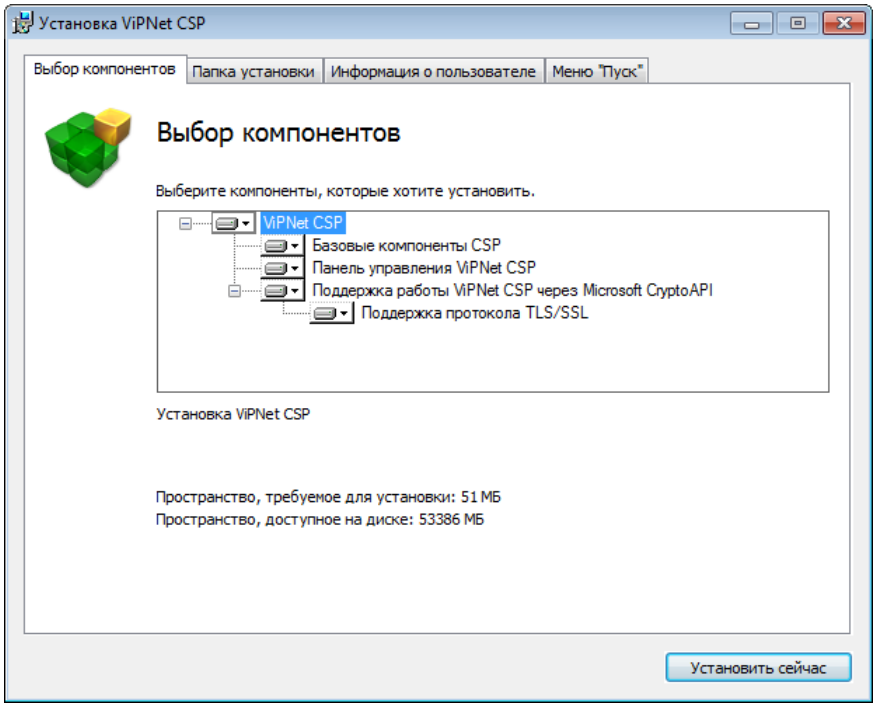

Рисунок 2. Настройка параметров установки ViPNet CSP

- Программа SoapUI.

  Установочный файл программы SoapUI вы можете бесплатно загрузить на официальном сайте проекта http://www.soapui.org/.

- Тестовая веб-страница (на стр. 11).

  Тестовая веб-страница содержится в файле `sign_demo.html` и поставляется в заархивированном виде `sign_demo.tar.gz`. Перед использованием извлеките файл `sign_demo.html` из архива на USB-носитель.

Получение сертификата и закрытого ключа

Чтобы создать запрос на сертификат и записать на криптотокен контейнер ключей, выполните следующие действия:

1. Подключите к компьютеру, на котором установлена программа ViPNet CSP, криптотокен.
2. Откройте в веб-браузере страницу для формирования запроса на сертификат в тестовом удостоверяющем центре ООО «КРИПТО-ПРО» http://www.cryptopro.ru/certsrv/certrqma.asp.
3. В форме запроса введите корректные данные и укажите следующие параметры:
   - В списке **CSP** выберите **Infotecs Cryptographic Service Provider**.
   - Установите переключатель **Использование ключей** в положение **Подпись**.
   - В списке **Алгоритм хеширования** выберите **ГОСТ**.

После заполнения запроса нажмите кнопку **Выдать**.

4 В появившемся окне введите ПИН-код криптотокена. В результате на криптотокене будет сформирована пара ключей (закрытый и открытый) и создан контейнер с закрытым ключом.

5 Согласитесь с установкой сертификата и в появившемся окне снова введите ПИН-код криптотокена. В результате в контейнер с закрытым ключом будет установлен сертификат открытого ключа, выданный тестовым удостоверяющим центром.

Чтобы проверить наличие контейнера ключей на криптотокене и его содержимое, выполните следующие действия:

1 Запустите программу ViPNet CSP.

2 В окне **ViPNet CSP** в разделе **Контейнеры ключей** выберите ваш криптотокен. Если на криптотокене есть контейнер ключей, он появится в списке.

3 Выберите в списке контейнер ключей и нажмите кнопку **Свойства**.

4 Убедитесь, что контейнер содержит закрытый ключ и сертификат открытого ключа.

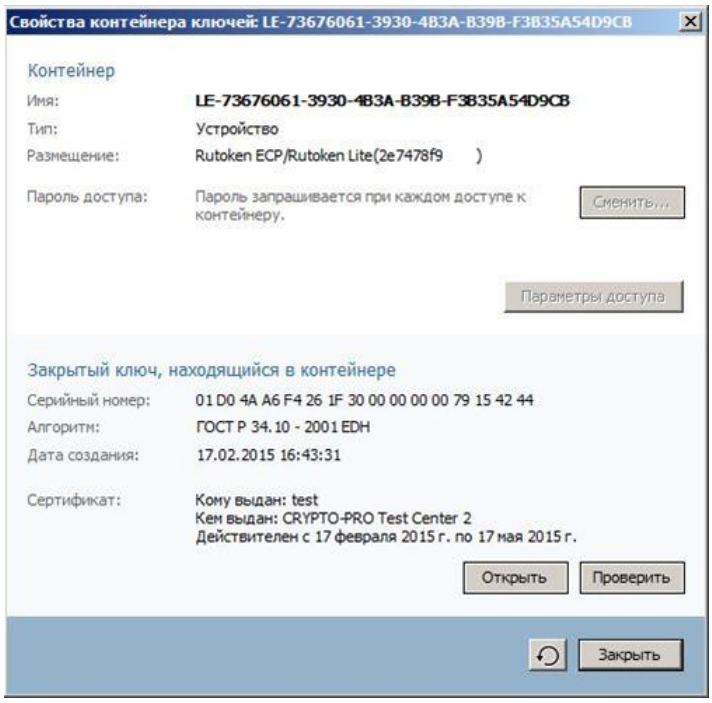

Рисунок 3. Свойства контейнера ключей

5 Нажмите кнопку **Открыть** и в окне свойств сертификата проверьте параметры сертификата. Если они соответствуют параметрам, которые вы задали в запросе на сертификат, криптотокен готов к использованию.

Формирование подписи

Чтобы сформировать подпись на сетевом узле ViPNet Terminal, выполните следующие действия:

1 Запустите ViPNet Terminal, на котором доступен терминальный сервер с подключением по протоколу HTTP, а также разрешено перенаправление устройств аутентификации.

2 Подключите к компьютеру USB-носитель с тестовой веб-страницей `sign_demo.html` (см. «Требования к дополнительному ПО» на стр. 7).

3 Подключитесь к терминальному серверу и в веб-браузере Firefox запустите тестовую веб-страницу, находящуюся на USB-носителе.

4 На веб-странице введите любые данные в поля ввода.

5 Если вы хотите добавить подпись к данным, установите флажок **Присоединить данные к подписи**. Если флажок не установлен, сформированная подпись будет сохранена в отдельном файле.

6 Подключите к компьютеру криптотокен.

7 В поле **ПИН-код** введите ПИН-код криптотокена.

8 В списке **Поддерживаемые устройства** выберите тип криптотокена, затем нажмите кнопку **Отобразить**. В списке **Сертификаты** появится записанный на криптотокене сертификат.

9 Выберите сертификат и нажмите кнопку **Подписать**. В поле **Данные (base64)** одной строкой будут отображены введенные данные, в поле **Подпись (base64)** — сформированная подпись. Если установлен флажок **Присоединить данные к подписи**, поле **Подпись (base64)** будет содержать подпись вместе с данными. Скопируйте строку подписи в буфер обмена.

10 Нажмите сочетание клавиш **Ctrl+Alt+F1**, чтобы перейти из графического режима в режим командного интерпретатора.

11 Выполните команду:

```
vpnshell echo <буфер обмена> >p7s.txt
```

Вместо `<буфер обмена>` вставьте в команду содержимое буфера обмена.

12 Подключите к компьютеру USB-носитель, подождите не менее 15 секунд, затем скопируйте на USB-носитель файл `p7s.txt` с помощью команды:

```
vpnshell cp p7s.txt /mnt/drive/A
```

13 Отключите USB-носитель с файлом подписи и используйте его для проверки подписи.

Проверка подписи

Чтобы проверить подпись, выполните следующие действия:

1 Подключите к компьютеру, на котором установлена программа SoapUI, USB-носитель с файлом `p7s.txt`.

2 Запустите программу SoapUI.

3 В главном меню выберите пункт **File** > **New soapUI Project**.

4 В окне **New soapUI Project** в поле **Project Name** введите любое имя проекта, в поле **Initial WSDL/WADL** введите адрес `http://195.245.214.33:7777/esv?wsdl`. Затем нажмите кнопку **OK**.

В результате на панели навигации появится проект с заданным именем, содержащий два веб-сервиса: VerifyCMSSignatureWithReport для проверки присоединенной подписи и

VerifyCMSSignatureDetachedWithReport для проверки отсоединенной подписи (сохраненной отдельно от данных). Каждый веб-сервис будет содержать автоматически созданный запрос Request1.

5 На панели навигации дважды щелкните запрос Request1 соответствующего веб-сервиса (в зависимости от того, какая подпись записана на USB-носитель).

6 В запросе укажите следующие значения:

- Для присоединенной подписи:
  - значение true: verifySignatureOnly>true;
  - message = подпись (base64).
- Для отсоединенной подписи:
  - значение true: verifySignatureOnly>true;
  - message = подпись (base64);
  - значение данные (base64) данные (base64).

7 Отправьте запрос Request1. Для этого в окне с запросом на панели инструментов нажмите зеленый треугольник.

После выполнения запроса появится сообщение с результатом проверки подписи — действительна или не действительна электронная подпись.

Тестовая веб-страница

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="ru_RU" xml:lang="ru_RU" >
<head>
<title>Demo page for local signature on ViPNet Terminal</title>
<meta http-equiv="Content-type" content="text/html; charset=utf-8"/>
<script type="text/javascript" src="sign_demo/soapclient21.js"></script>
<script type="text/javascript" src="sign_demo/validate.js"></script>
<script type="text/javascript" src="sign_demo/sign.js"></script>

<script type="text/javascript">
function getProp ()
{
  // The first way to work with plugin
  var o = document.getElementById("signer");
  if(o) {
    var v = o.propName;
    document.getElementById("propValue").value = v;
  }
}

function callFunc ()
{
```

```
  // The second way to work with plugin dynamically instantiating object
  var mimetype = "application/x-itcs-digitalsign";
  var mt = navigator.mimeTypes[mimetype];
  if(mt) {
    var o = document.createElement("embed");
    o.setAttribute("type", mimetype);
    o.setAttribute("hidden", true);
    document.body.appendChild(o);
    var v = o.funcName();
    document.getElementById("funcValue").value = v;
  }
}
</script>
</head>

<body>
<!-- The second way to include plugin -->
<embed id="signer" type="application/x-itcs-digitalsign" hidden="true">
<h1>Пример локальной подписи на ViPNet Terminal</h1>
<p><b>Форма для ввода пользовательских данных</b></p>
<div style="position: absolute; width: 450px; height: 1100px; z-index: 2; left: 900px;
top: 0px; background-color: #EAEAEA" id="layer2">
  <div style="position: absolute; width: 397px; height: 124px; z-index: 1; left: 7px;
  top: 125px" id="layer3">
    <p>
      <font face="Verdana" size="2" color="#000080">
        <textarea style="font-size: 12" rows="22" name="Certs" id="Certs" cols="51">
        </textarea>
      </font>
    </p>
  </div>

  <div style="position: absolute; z-index: 1; left: 7px; top: 485px" id="Cert">
    <p>
      <font face="Verdana" size="2" color="#000080">
        <input value="Отобразить" onclick="getCertList()" type="button">
        </input>
      </font>
    </p>
  </div>

  <div style="position: absolute; z-index: 1; left: 7px; top: 60px" id="Div1">
    <select name="selectToken" class="small" onchange="selectToken()">
      <option value="0" selected="selected" >Поддерживаемые устройства</option>
      <option value="1">eToken GOST</option>
      <option value="2">Rutoken ECP</option>
    </select>
  </div>

  <div style="position: absolute; z-index: 1; left: 7px; top: 90px" id="Div2">
    <p>
```

```
      <b>
        <font face="Verdana"><span lang="ru"> Сертификаты</span>
        </font>
      </b>
    </p>
  </div>
</div>

<table border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td>Поле ввода данных #1</td>
    <td><input id="dataField1" maxlength="100" size = "100" name="field1"></td>
  </tr>
  <tr>
    <td>Поле ввода данных #2</td>
    <td><input id="dataField2" maxlength="100" size = "100" name="field2"></td>
  </tr>
  <tr>
    <td align="left">ПИН-код</td>
    <td><input id="pin" maxlength="10" type="password"></td>
  </tr>
  <tr>
    <td align="left">Присоединить данные к подписи</td>
    <td><input type="checkbox" name="att" value="qqq" id="att"></td>
  </tr>
  <tr>
    <td colspan="2"><input value="Подписать" onclick="signData()" type="button"></td>
  </tr>
  <tr>
    <td>Данные (base64):</td>
    <td>
      <p style="margin-top: 0; margin-bottom: 0">
        <font face="Verdana" size="2" color="#000080">
          <textarea style="font-size: 12" rows="7" name="S12" id="data" cols="75">
          </textarea>
        </font>
      </p>
    </td>
  </tr>
  <tr>
    <td>Подпись (base64):</td>
    <td>
      <p style="margin-top: 0; margin-bottom: 0">
        <font face="Verdana" size="2" color="#000080">
          <textarea style="font-size: 12" rows="25" name="S12" id="signature"
          cols="75">
          </textarea>
        </font>
      </p>
    </td>
  </tr>
<!--
```

```
  <tr>
    <td colspan="2"><input value="Проверить подпись" onclick="validateSignature()"
    type="button"></td>
-->
  </tr>
</table>

<h1>Тест функций плагина</h1>
<p>Форма для ввода/вывода тестовых данных</p>
  <table border="0" cellpadding="0" cellspacing="0">
    <tr>
      <td colspan="2"><input value="Свойство" onclick="getProp()" type="button"></td>
    </tr>
    <tr>
      <td>Результат</td>
      <td><input id="propValue" maxlength="30" name="propValue"></td>
    </tr>
    <tr>
      <td>
        <br>
      </td>
    </tr>
    <tr>
      <td colspan="2"><input value="Функция" onclick="callFunc()" type="button"></td>
    </tr>
    <tr>
      <td>Результат</td>
      <td><input id="funcValue" maxlength="30" name="funcValue"></td>
    </tr>
  </table>
</body>
</html>
```