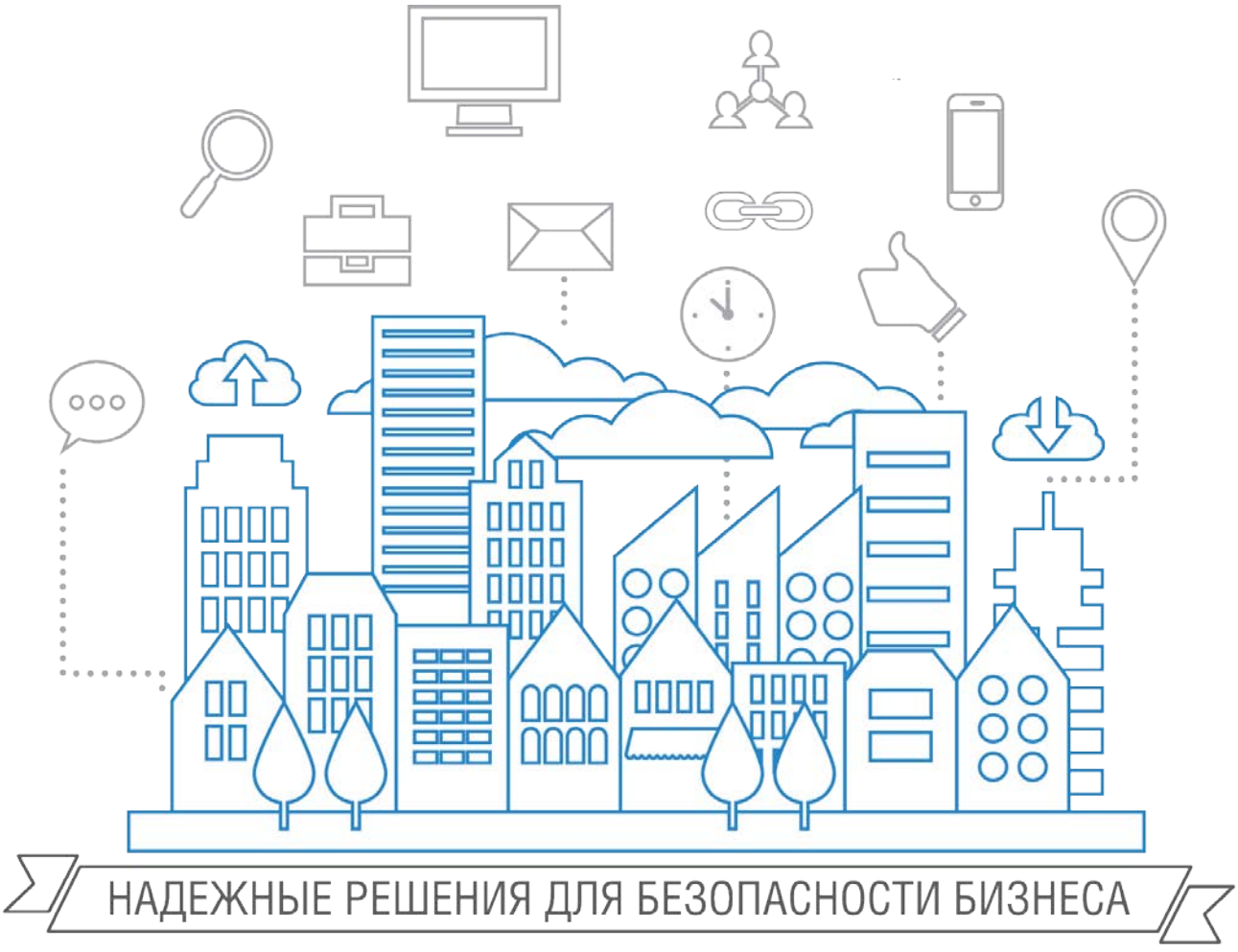
ГАЗИНФОРМСЕРВИС

198096, г. Санкт-Петербург, ул. Кронштадтская, д.10, лит. А, тел.: (812) 677-20-50, факс: (812) 677-20-51
Почтовый адрес: 198096, г. Санкт-Петербург, а/я 59, e-mail: resp@gaz-is.ru, www.gaz-is.ru
р/с 40702810800000001703 Ф-л Банка ГПБ (АО) в г. Санкт-Петербурге БИК 044030827,
к/с 30101810200000000827, ОКПО 72410666, ОГРН 1047833006099, ИНН/КПП 7838017968/783450001

Библиотека «GiSCryptoNET» (Demo-версия)

Описание интерфейсов



Санкт-Петербург, 2017

Аннотация

В документе приводится описание программных интерфейсов библиотеки «GiSCryptoNET» (Demo-версия), реализованных на языке C#, с помощью которых вызываются основные функции криптографической платформы «Litoria Crypto Platform».

В разделе «Общие сведения» описано назначение библиотеки «GiSCryptoNET» и указаны функции, которые вошли в Demo-версию библиотеки.

В разделе «Описание программных интерфейсов» приведена классификация программных интерфейсов, а также указаны заголовки объявления, входные параметры, возвращаемые значения и примеры вызовов интерфейсов.

В разделе «Сообщения об ошибках» описан метод обработки исключительных ситуаций и указаны коды ошибок.

В конце документа приведен список использованных терминов и сокращений.

Содержание

1 Общие сведения

Библиотека «GiSCryptoNET» (Demo-версия) содержит программные интерфейсы, реализованные на языке верхнего уровня C#, и предназначена для вызова основных функций криптографической платформы «Litoria Crypto Platform».

В библиотеку «GiSCryptoNET» (Demo-версия) включены интерфейсы для вызова следующих функций криптографической платформы «Litoria Crypto Platform»:

- получение коллекции сертификатов из хранилища;
- установка сертификата в хранилище (без привязки к закрытому ключу);
- создание усовершенствованной электронной подписи (УЭП);
- добавление УЭП;
- проверка УЭП;
- шифрование файла;
- расшифровывание файла.

2 Описание программных интерфейсов

При использовании программных интерфейсов библиотеки «GiSCryptoNET» (Demo-версия) и наличии на рабочей станции криптографической платформы «Litoria Crypto Platform» можно реализовать клиентское программное обеспечение, в котором пользователю предоставляются перечисленные выше функции криптографической платформы «Litoria Crypto Platform».

Классификация программных интерфейсов представлена в таблице 2.1.

Таблица 2.1. Классификация программных интерфейсов

Тип интерфейсов	Интерфейс	Назначение
Интерфейс для настройки доступа к прокси-серверу	SetProxy	Настройка имени пользователя и пароля для доступа к прокси-серверу
Интерфейсы для работы с сертификатами	GetAllCertFromStore	Получение коллекции сертификатов из хранилища
	InstallCert	Установка сертификата в заданное хранилище (без привязки к закрытому ключу)
Интерфейсы для работы с ЭП	SignData	Создание УЭП
	AddSign	Добавление УЭП
	VerifySignature	Проверка УЭП
Интерфейс для шифрования файла	EncryptData	Шифрование файла
Интерфейс для расшифровывания файла	DecryptData	Расшифровывание файла
Интерфейс для обработки ошибок	GetBHGiSErrorInfoString	Получение описания ошибки криптографической платформы «Litoria Crypto Platform»

2.1 Доступ к прокси-серверу. Интерфейс SetProxy

Заголовок объявления интерфейса:
public static void SetProxy(string name, string pass)

Входные параметры:
name – имя пользователя
pass – пароль

Пример вызова:
GiSCryptoNET.GiSCryptoDemo.SetProxy("domen\\user", "123456");

2.2 Получение коллекции сертификатов из хранилища. Интерфейс GetAllCertFromStore

Заголовок объявления интерфейса:
public static X509Certificate2Collection GetAllCertFromStore(string storeName)

Входные параметры:
StoreName – имя хранилища
Примеры имен хранилищ:
- «MY» – хранилище личных сертификатов
- «CA» – сертификаты удостоверяющего центра
- «ROOT» – корневые сертификаты
- «AddressBook» – сертификаты других пользователей

Возвращаемое значение:

Коллекция сертификатов

Пример вызова:

```
X509Certificate2Collection x509Certs2 =
GiSCryptoNET.GiSCryptoDemo.GetAllCertFromStore("AddressBook");
```

2.3 Установка сертификата в хранилище. Интерфейс InstallCert

Заголовок объявления интерфейса:

```
public static void InstallCert(X509Certificate2 certForInstall, string storeName)
```

Входные параметры:

certForInstall – устанавливаемый сертификат

storeName – имя хранилища

Примеры имен хранилищ:

- «MY» – хранилище личных сертификатов
- «CA» – сертификаты удостоверяющего центра
- «ROOT» – корневые сертификаты
- «AddressBook» – сертификаты других пользователей

Пример вызова:

```
var inputStream = new System.IO.FileStream("C:\\CertFile.cer", System.IO.FileMode.Open);
var certRawData = new Byte[inputStream.Length];
inputStream.Read(certRawData, 0, certRawData.Length);
inputStream.Close();
var  x509Cert2 = new X509Certificate2(certRawData);
GiSCryptoNET.GiSCryptoDemo.InstallCert(x509Cert2, "AddressBook");
```

2.4 Создание УЭП. Интерфейс SignData

Заголовок объявления интерфейса:

```
public static void SignData(X509Certificate2 certificate, string inFileName, string tspUrl, string outputFileName)
```

Входные параметры:

certificate – сертификат подписывающего лица

inFileName – имя подписываемого файла

tspUrl – адрес используемой службы штампов времени

outputFileName – имя результирующего файла

Пример вызова:

```
var x509Certs2 = GiSCryptoNET.GiSCryptoDemo.GetAllCertFromStore("MY");
GiSCryptoNET.GiSCryptoDemo.SignData(x509Certs2[0],  "C:\\test.txt",  "http://tsp.gaz-is.ru/tsp/tsp.srf",
"C:\\test.txt.p7s");
```

2.5 Добавление УЭП. Интерфейс AddSign

Заголовок объявления интерфейса:

```
Public static void AddSign(X509Certificate2 certificate, string inFileName, string tspUrl)
```

Входные параметры:

certificate – сертификат подписывающего лица

inFileName – имя подписанного файла, к которому добавляется подпись

tspUrl – адрес используемой службы штампов времени

Пример вызова:
var x509Certs2 = GiSCryptoNET.GiSCryptoDemo.GetAllCertFromStore("MY");
GiSCryptoNET.GiSCryptoDemo.AddSign(x509Certs2[0], "C:\\test.txt.p7s", "http://tsp.gaz-is.ru/tsp/tsp.srf");

2.6 Проверка УЭП. Интерфейс VerifySignature

Заголовок объявления интерфейса:
public static SignatureInfo[] VerifySignature(string inFileName, bool getData, string outputFileName)

Входные параметры:
inFileName – имя проверяемого подписанного файла
GetData – флаг снятия УЭП (если true, на выходе получаем исходные данные)
outputFileName – имя результирующего файла (в случае, если getData = true)

Возвращаемое значение:
Информация о проверке УЭП (SignatureInfo[]):
int *index* – номер подписи
bool *verifyResult* – результат проверки математической целостности подписи
bool *verifyCertificateResult* – результат проверки сертификата подписчика
X509Certificate2 *certificate* – сертификат подписчика
DateTime *signatureDate* – дата создания подписи
byte[] *rawSignature* – оттиск подписи
string *comment* – комментарий к подписи
EDSException *signatureException* – возникшие ошибки при проверке

Пример вызова:
SignatureInfo[] signatureInfo;
signatureInfo = GiSCryptoNET.GiSCryptoDemo.VerifySignature("C:\\test.txt.p7s", true, "C:\\test.txt");

2.7 Шифрование файла. Интерфейс EncryptData

Заголовок объявления интерфейса:
public static void EncryptData(X509Certificate2Collection certificates, string inFileName, string outputFileName)

Входные параметры:
certificates – сертификаты получателей
inFileName – имя файла, который необходимо зашифровать
outputFileName – имя результирующего файла

Пример вызова:
var x509Certs2 = GiSCryptoNET.GiSCryptoDemo.GetAllCertFromStore("AddressBook ");
GiSCryptoNET.GiSCryptoDemo.EncryptData(x509Certs2, "C:\\test.txt", "C:\\test.txt.p7m");

2.8 Расшифрование файла. Интерфейс DecryptData

Заголовок объявления интерфейса:
public static X509Certificate2 DecryptData(string inFileName, string outputFileName)

Входные параметры:
inFileName – имя файла, который необходимо расшифровать
outputFileName – имя результирующего файла

Выходные параметры:
Сертификат получателя зашифрованного сообщения

Пример вызова:
var x509ReceiverCert = GiSCryptoNET.GiSCryptoDemo.DecryptData("C:\\test.txt.p7m", "C:\\test.txt");

2.9 Получение описания ошибки криптографической платформы «Litoria Crypto Platform». Интерфейс GetBHGiSErrorInfoString

Заголовок объявления интерфейса:
public static string GetBHGiSErrorInfoString(int bhGiSErrorCode)

Входные параметры:
bhGiSErrorCode – код ошибки криптографической платформы «Litoria Crypto Platform»

Выходные параметры:
Текстовое описание ошибки криптографической платформы «Litoria Crypto Platform»

Пример вызова:
Смотрите раздел 3.

3 Описание программных интерфейсов C#

Функции обработки исключений помогают обрабатывать любые непредвиденные или исключительные ситуации, происходящие при выполнении основных операций. Для обработки исключений в программных интерфейсах используются ключевые слова try и catch.

Пример обработки исключения при выполнении операции создания УЭП:

```
private void btConvert_Click(object sender, EventArgs e)
      {
         try
         {
         var x509Certs2 = GiSCryptoNET.GiSCryptoDemo.GetAllCertFromStore("MY");
         GiSCryptoNET.GiSCryptoDemo.SignData(x509Certs2[0], "C:\\test.txt", "http://tsp.gaz-
is.ru/tsp/tsp.srf", "C:\\test.txt.p7s");
         }
         catch (GiSCryptoNET.GiSCryptoDemo.EDSException ex)
         {
            MessageBox.Show(ex.systemErrorCode.ToString());
            MessageBox.Show(ex.systemErrorText);
            for (int i = 0; i < ex.bhGiSErrorCode.Length; i++)
          {
                var bhErrorText =
         GiSCryptoNET.GiSCryptoDemo.GetBHGiSErrorInfoString(ex.bhGiSErrorCode[i]);
                MessageBox.Show(bhErrorText);
 }
         }
      }
```

Описание параметров, содержащих информацию об ошибке, представлено в таблице 3.1

Таблица 3.1. Параметры, содержащие информацию об ошибке

Параметр	Значение
systemErrorCode	Код ошибки, соответствующей таблице кодов ошибок Microsoft Windows.
systemErrorText	Текст ошибки, соответствующей таблице кодов ошибок Microsoft Windows.
bhGiSErrorCode	Коды ошибок криптографической платформы «Litoria Crypto Platform».

Термины и сокращения

CA	–	Certifying Authority (Сертифицирующая организация)
TSP	–	Time-Stamp Protocol (Протокол штампов времени)
Сертификат	–	документ на бумажном носителе или электронный документ с электронной подписью уполномоченного лица удостоверяющего центра, которые включают в себя открытый ключ ЭП, и которые выдаются удостоверяющим центром участнику информационной системы для подтверждения подлинности ЭП и идентификации владельца сертификата ключа подписи.
Усовершенствованная электронная подпись (УЭП)	–	электронная подпись, усовершенствованная, в качестве меры борьбы с общепризнанными угрозами безопасности, добавлением (как необходимое требование) признаков ее (подписи) регламента и доказательств подлинности таких, как штамп времени, данные об отзыве сертификата и др.
Электронная подпись (ЭП)	–	реквизит электронного документа, предназначенный для защиты данного электронного документа от подделки, полученный в результате криптографического преобразования информации с использованием закрытого ключа ЭП и позволяющий идентифицировать владельца сертификата ключа подписи, а также установить отсутствие искажения информации в электронном документе.