**// 1. Подписание и шифрование**

МассивЗащищенныхФайлов = EUTaxService\_ProtectFilesEx(МассивФайловДляЗащиты, МассивПодписей);

Функция **EUTaxService\_ProtectFilesEx**(МассивФайлов, МассивПодписей = Неопределено, РасширениеВыходныхФайлов = "p7s") Экспорт

МассивВыходныхФайлов = Новый Массив;

МассивПодписей = Новый Массив;

ЕстьОшибка = Ложь;

// Создаем массив сертификатов для подписания

МассивСертификатов = Новый Массив;

Если НЕ ЗначениеЗаполнено(EUTaxService\_СертификатМДС) Тогда

Сообщить("Не выбран сертификат МДС!", СтатусСообщения.Важное);

Возврат Неопределено;

Иначе

МассивСертификатов.Добавить(EUTaxService\_СертификатМДС);

КонецЕсли;

Если НЕ ЗначениеЗаполнено(EUTaxService\_СертификатExite) Тогда

Сообщить("Не выбран сертификат Exite!", СтатусСообщения.Важное);

Возврат Неопределено;

Иначе

МассивСертификатов.Добавить(EUTaxService\_СертификатExite);

КонецЕсли;

Попытка

//Вводим ключи

Если НЕ EUTaxService\_ВвестиКлючи(euKeyTypeAccountant, EUTaxService\_ФайлКлючаБухгалтера, EUTaxService\_ПарольКлючаБухгалтера) Тогда

Возврат Неопределено;

КонецЕсли;

Если НЕ EUTaxService\_ВвестиКлючи(euKeyTypeDigitalStamp, EUTaxService\_ФайлКлючаПечати, EUTaxService\_ПарольКлючаПечати) Тогда

Возврат Неопределено;

КонецЕсли;

serverCertIDs = Новый COMSafeArray(МассивСертификатов, "VT\_BSTR");

// Метод встановлення необхідності використання транспортного та криптозаголовків

// Розповсюджується на всі функції. Якщо встановлений в FALSE ігнорує параметри

// використання транспортного та криптозаголовків, що передаються в функцію

// useHeaders - параметр, що передається. Містить інформацію про

// признак необхідності використання транспортного та

// криптозаголовків

useHeaders = Истина;

EUTaxService\_Объект.SetFilesOptions(useHeaders);

// Метод захисту файлів з додатковими параметрами. Захист файлів передбачає виконання

// наступних криптографічних операцій (у порядку їх виконання):

// 1) підпис за допомогою ключа бухгалтера(якщо useAccountant = TRUE);

// 2) підпис за допомогою ключа директора(якщо useDirector = TRUE);

// 3) підпис за допомогою ключа цифрової печатки (якщо useDigitalStamp = TRUE);

// 4) направлене шифрування з використанням сертифіката сервера одержувача, та

// особистого ключа цифрової печатки (якщо useDigitalStamp = TRUE, значення

// useDirector в цьому випадку ігнорується) або директора

// 5) підпис зашифрованих даних за допомогою ключа цифрової печатки (якщо

// useDigitalStamp = TRUE, значення useDirector в цьому випадку ігнорується) або

// директора

МассивФайловCFA = Новый COMSafeArray(МассивФайлов, "VT\_BSTR");

useAccountant = Истина;

useDirector = Ложь;

useDigitalStamp = Истина;

appendCryptoHeaders = Истина;

appendTransportHeader = Ложь;

emailAddress = "pn@exite.ua";

ProtectFilesResultsCFA = EUTaxService\_Объект.ProtectFilesEx(МассивФайловCFA, useAccountant, useDirector, useDigitalStamp, appendCryptoHeaders, appendTransportHeader, emailAddress, serverCertIDs);

ProtectFilesResults = ProtectFilesResultsCFA.Выгрузить();

Для каждого ЭлементМассива Из ProtectFilesResults Цикл

Если ЭлементМассива.IsSuccessfull() Тогда

// Переименуем созданный файл

ИсходныйФайл = Новый Файл(ЭлементМассива.GetInputFileName());

НовыйФайл = Новый Файл(ЭлементМассива.GetOutputFileName());

НовоеИмяФайла = НовыйФайл.Путь + ИсходныйФайл.ИмяБезРасширения + "." + РасширениеВыходныхФайлов;

ПереместитьФайл(НовыйФайл.ПолноеИмя, НовоеИмяФайла);

МассивВыходныхФайлов.Добавить(НовоеИмяФайла);

Сообщить(ИсходныйФайл.ПолноеИмя + " => " + НовоеИмяФайла);

//УдалитьФайлы(ИсходныйФайл.ПолноеИмя);

МассивПодписей.Добавить(Новый Структура("ИмяФайла, Подписи", ИсходныйФайл.ПолноеИмя, ПолучитьСтруктуруИнформацииИзРезультата(ЭлементМассива)));

Иначе

Сообщить("Ошибка защиты файла " + ЭлементМассива.GetInputFileName() + " по причине: " + ЭлементМассива.GetStatusDescription());

ЕстьОшибка = Истина;

КонецЕсли;

КонецЦикла;

Исключение

Сообщить("Не удалось выполнить функции защиты файлов" +

Символы.ПС + ОписаниеОшибки() +

Символы.ПС + EUTaxService\_Объект.GetLastErrorDescription(), СтатусСообщения.Важное);

Возврат Неопределено;

КонецПопытки;

Если ЕстьОшибка Тогда

Возврат Неопределено;

КонецЕсли;

Возврат МассивВыходныхФайлов;

КонецФункции

--------------------------------------------------------------------------------

**// 2. Отправить на FTP**

ОтправитьНаFTP(МассивЗащищенныхФайлов);

Процедура **ОтправитьНаFTP**(МассивФайлов)

Сервер = ПолучитьFTPСоединение();

Если Сервер = Неопределено Тогда

Сообщить("Отсутствует подключение к FTP! (335)", СтатусСообщения.Важное);

Возврат;

КонецЕсли;

Для каждого ЭлементМассивФайлов Из МассивФайлов Цикл

Файл = Новый Файл(ПолноеИмяФайла);

Если Файл.Существует() Тогда

Сервер.Записать(ПолноеИмяФайла, "/outbox/" + Файл.Имя);

КонецЕсли;

КонецЦикла;

КонецПроцедуры