DOI: 10.24888/2500-1957-2020-4-85-93

УДК 004.422.8

ПРИМЕРЫ РЕАЛИЗАЦИИ МЕХАНИЗМА КРИПТОГРАФИИ В СИСТЕМЕ 1С:ПРЕДПРИЯТИЕ 8

Светлана Викторовна Мишина
ст. преподаватель
svmishina2017@mail.ru
г. Елец

Елецкий государственный университет им. И. А. Бунина

Аннотация. Для современного предприятия скорость бизнес-процессов – это одно из конкурентных преимуществ. При применении электронного документооборота затраты времени уменьшаются минимум на 75%. Помимо этого, происходит экономия собственных ресурсов (бумаги, картриджей для принтеров, места для хранения документов, времени сотрудников на организацию бумажного документооборота). Все это сильно помогает предприятию. Почти каждое предприятие сейчас использует дистанционные каналы банковского обслуживания, и безопасность проведения удаленных платежей на текущий момент весьма актуальна. Все крупные налогоплательщики сейчас включают в договор тезисы о том, что, если Вы хотите с ними работать – обменивайтесь документами в электронном виде. Все организации постепенно переходят на электронные счета-фактуры, электронные заказы и т. д. А если хотите участвовать в торгах – у Вас должна быть электронная подпись. Все это постепенно становится массовым. Но, несмотря на всю простоту использования данных возможностей, есть факторы, которые негативно влияют на развитие электронного документооборота. Это связано со сложностью защиты данных, которые участвуют в данном процессе. При этом преследуются две цели: обеспечивается конфиденциальность личной и деловой информации, а также гарантируется точность хранимой информации. Для их реализации в системе 1С:Предприятие 8 реализован механизм криптографии. Описанию способов его использования и посвящена данная статья.

Ключевые слова: шифрование, криптография, цифровая подпись, открытый ключ, закрытый ключ, сертификат.

При использовании системы «1С:Предприятие» в системах автоматизации может быть необходимо обеспечить проверку, что тот или иной документ, хранящийся в системе, не был изменен (например, к объекту базы данных Договор прикладывается документ, содержащий текст самого договора). Также может возникать необходимость обеспечить передачу какой-либо подписанной информации или организовать утверждение какого-либо документа в рамках системы. Возможны ситуации, когда требуется обеспечить передачу информации по открытым каналам так, чтобы с ней было невозможно ознакомиться, перехватив передачу (зашифровать данные). Для подобных ситуаций в системе «1С:Предприятие» реализован механизм криптографии, базирующийся на асимметричном шифровании (используется пара ключей – открытый и закрытый). Механизм криптографии «1С:Предприятия» не содержит реализации собственно алгоритмов криптографии. Он обеспечивает набор объектов, позволяющих взаимодействовать с внешними модулями криптографии сторонних производителей. Открытый ключ предназначен для передачи по открытым каналам. Закрытый ключ для распространения не предназначен и должен быть максимально защищен. Чтобы

зашифровать данные, необходимо знать открытые ключи получателей. Для расшифровки нужно иметь закрытый ключ, являющийся парой для открытого, указанного при осуществлении шифрования. Для генерации цифровой подписи нужен закрытый ключ, а для проверки подписи – открытый ключ подписавшего (чаще всего открытый ключ включен в саму подпись).

При шифровании данных, кроме асимметричного шифрования, используется симметричное шифрование. Симметричное шифрование – это способ шифрования, когда для шифрования и расшифровки используется один и тот же ключ. Для того чтобы подтвердить, что конкретный открытый ключ действительно принадлежит некоторому субъекту, служит удостоверяющий центр, который предоставляет такую гарантию посредством подписи. Сертификат – это открытый ключ, подписанный удостоверяющим центром. Ввиду необходимости защиты закрытый ключ существует в контейнере. Контейнер ключей может включать помимо закрытого ключа открытый ключ (например, в виде сертификата).

При работе с механизмом криптографии следует учитывать, где выполняется эта работа. Причина заключается в том, что схемы работы с криптографией на стороне клиентского приложения и на стороне сервера существенно отличаются. На стороне сервера используются так называемые синхронные методы работы с криптографией, в то время как на стороне клиентского приложения используются асинхронные методы работы. Для работы в веб-клиенте должно быть установлено расширение работы с криптографией. Особенностью веб-клиента является выполнение запроса разрешения пользователя на выполнение некоторых операций: доступ к файловой системе и к закрытому ключу.

При работе с криптографическими механизмами можно выделить два основных варианта использования: шифрование/расшифровка данных; подпись/проверка подписи данных.

Общая схема работы при шифровании/расшифровке выглядит следующим образом:

1. Создается необходимый объект доступа к криптографической функциональности (объект МенеджерКриптографии).
2. Выбирается необходимый сертификат, который является открытым ключом принимающей стороны (объект СертификатКриптографии). С помощью этого сертификата будет выполняться шифрование данных.
3. Выполняется шифрование необходимого файла или двоичных данных с помощью методов Зашифровать() созданного объекта типа МенеджерКриптографии. В процессе шифрования используется комбинация симметричного и асимметричного шифрования. Вначале система формирует ключ для симметричного шифрования, затем этим ключом выполняется шифрование требуемых данных. Затем ключ симметричного шифрования шифруется асимметричным ключом. Пакет данных (по спецификации PKCS#7), пригодный для передачи по открытым каналам, формируется из трех составляющих: зашифрованные данные, зашифрованный ключ для симметричного шифрования и список получателей.
4. Зашифрованные данные готовы для передачи по открытым каналам.
5. При получении зашифрованных данных выполняется обратная операция.
6. Создается необходимый объект доступа к криптографической функциональности. Модуль криптографии, связанный с объектом, должен поддерживает работу с алгоритмами, которые использованы для шифрования данных.
7. Выполняется расшифровка полученных данных с помощью метода Расшифровать() созданного объекта доступа к криптографической функциональности.

При расшифровке из пришедшего пакета извлекается зашифрованный ключ для симметричного шифрования и расшифровывается с использованием закрытого ключа (парный к открытому ключу, который использовался для шифрования). Затем полученный ключ используется в симметричном алгоритме для расшифровки массива полученных

данных. Таким образом, если принимающая сторона («из» асимметричного шифрования) не имеет закрытого ключа, будет невозможно расшифровать данные симметричным алгоритмом.

Общая схема работы при подписи/проверке подписи выглядит следующим образом:

1. Создается необходимый объект доступа к криптографической функциональности (объект МенеджерКриптографии).
2. Выбирается необходимый сертификат, связанный с закрытым ключом подписывающей стороны (объект СертификатКриптографии). С помощью этого сертификата будет выполняться формирование ЭЦП.
3. Выполняется формирование ЭЦП необходимого файла или двоичных данных с помощью методов Подписать() созданного объекта типа МенеджерКриптографии.

При формировании подписи следует учитывать следующие особенности: вначале формируется хеш-сумма подписываемых данных и затем выполняется подпись полученной хеш-суммы. Это сделано для того, чтобы уменьшить время, необходимое для выполнения операции подписи данных.

В силу этого рекомендуется явным образом указывать алгоритмы, которые будут использоваться как для вычисления хеш-суммы (свойство АлгоритмХеширования), так и для выполнения подписи (свойство АлгоритмПодписи). Если принимающая сторона не будет поддерживать используемые алгоритмы (хеширования и подписи), то проверить подпись будет невозможно.

Для того чтобы явным образом указать используемые алгоритмы, следует воспользоваться методом ПолучитьИнформацияМодуляКриптографии() менеджера криптографии. В результате использования метода будет получен объект ИнформацияМодуляКриптографии, в котором свойства АлгоритмыПодписи и АлгоритмыХеширования будут содержать список алгоритмов, которые поддерживает используемый модуль криптографии. В свойства менеджера криптографии следует указывать те данные, которые содержатся в свойствах объекта ИнформацияМодуляКриптографии.

Информация об используемых алгоритмах будет содержаться в подписи, поэтому нет необходимости дополнительно передавать ее вместе с подписанными данными.

4. Подписываемые данные с электронной цифровой подписью передаются принимаемой стороне.
5. После получения подписанных данных и ЭЦП выполняется обратная операция.
6. Создается необходимый объект доступа к криптографической функциональности. Модуль криптографии, связанный с объектом, должен поддерживает работу с алгоритмами, которые использованы для формирования ЭЦП.
7. Выполняется проверка электронной подписи с помощью метода ПроверитьПодпись() созданного объекта доступа к криптографическому функционалу. Если модуль криптографии, используемый для проверки подписи, не поддерживает алгоритм, используемые для вычисления хеш-суммы и подписи данных, то проверить подпись будет невозможно и будет выдана ошибка.

Сертификаты, необходимые для выполнения криптографических операций, получают у соответствующего удостоверяющего центра.

При работе с объектом МенеджерКриптографии не рекомендуется получать несколько хранилищ сертификатов с одинаковыми характеристиками, поскольку модификация одного хранилища может привести к разному поведению другого, в зависимости от используемых модулей криптографии.

Для взаимодействия с модулями криптографии в ОС Windows используется Microsoft CryptoAPI. Для взаимодействия с модулями криптографии в ОС Linux используется непосредственное взаимодействие с установленными компонентами. А именно: средство криптографической защиты информации КриптоПро. Для использования данного средства следует в качестве параметра ТипМодуляКриптографии конструктора объекта МенеджерКриптографии указать значение 75.

Приведем примеры реализация некоторых типовых задач при использовании механизма криптографии.

Создание объекта доступа (листинг 1) к криптографическому функционалу является базовой операцией, без выполнения которой невозможны дальнейшие операции с механизмом криптографии.

```
МенеджерКриптографии = Новый МенеджерКриптографии("", "", 75);
```

Листинг 1. Создание объекта доступа

Формируется список сертификатов (листинг 2) из выбранных хранилищ сертификатов для дальнейшей работы с этим списком.

```
&НаКлиенте
Функция ПолучитьСписокСертификатов(ТипМенеджераКриптографии,
МассивТипов, ПроверятьДатуОкончания)
    СписокСертификатов = Новый Массив;
    МенеджерКриптографии = Новый МенеджерКриптографии("", "",
ТипМенеджераКриптографии);
    Для Каждого ТипХранилища Из МассивТипов Цикл
        // Получаем сертификаты для каждого типа хранилища сертификатов
        Хранилище =
МенеджерКриптографии.ПолучитьХранилищеСертификатов(ТипХранилища);
        СертификатыХранилища = Хранилище.ПолучитьВсе();
        ТекущаяДата = ТекущаяДата();
        Для Каждого Сертификат Из СертификатыХранилища Цикл
            Если ПроверятьДатуОкончания И Сертификат.ДатаОкончания <
ТекущаяДата Тогда
                // Пропускаем истекшие сертификаты, если нужно
                Продолжить;
            КонецЕсли;
            СписокСертификатов.Добавить(Сертификат);
        КонецЦикла;
    КонецЦикла;
    Возврат СписокСертификатов;
КонецФункции
&НаКлиенте
Процедура ПолучениеСпискаСертификатов()
    ТипыХранилищ = Новый Массив;
ТипыХранилищ.Добавить(ТипХранилищаСертификатовКриптографии.ПерсональныеС
ертификаты);
ТипыХранилищ.Добавить(ТипХранилищаСертификатовКриптографии.СертификатыПо
лучателей);
    Список = ПолучитьСписокСертификатов(75, ТипыХранилищ, Истина);
    // ...
КонецПроцедуры
```

Листинг 2. Получение списка сертификатов

Выполняется шифрование интерактивно выбранного файла и интерактивная запись зашифрованного файла на диск клиентского компьютера (листинг 3). В целях демонстрации для операции шифрования всегда используется первый по списку сертификат из всех сертификатов, установленных на компьютере.

```
&НаСервере
Функция ЗашифроватьНаСервере(АдресДанных, ДанныеСертификатов)
    // Создадим сертификаты на основании двоичных данных сертификатов с
клиента
    Сертификаты = Новый Массив();
    Для Каждого ДанныеСертификата Из ДанныеСертификатов Цикл
```

```
        Сертификаты.Добавить(Новый
СертификатКриптографии(ДанныеСертификата));
    КонецЦикла;
    МенеджерКриптографии = Новый МенеджерКриптографии("", "", 75);
    // Получим файл для шифрования из временного хранилища
    Данные = ПолучитьИзВременногоХранилища(АдресДанных);
    Если ТипЗнч(Данные) <> Тип("ДвоичныеДанные") Тогда
        Возврат Ложь;
    КонецЕсли;
    // Шифруем двоичные данные
    ЗашифрованныеДвоичныеДанные =
МенеджерКриптографии.Зашифровать(Данные, Сертификаты);
    // Сохраняем во временное хранилище
    АдресДанных =
ПоместитьВоВременноеХранилище(ЗашифрованныеДвоичныеДанные);
    Возврат Истина;
КонецФункции
&НаКлиенте
Процедура ШифрованиеФайла()
    Адрес = "";
    Результат = ПоместитьФайл(Адрес, , , Истина);
    Если Не Результат Тогда
        Возврат;
    КонецЕсли;
    ТипыСертификатов = Новый Массив;
ТипыСертификатов.Добавить(ТипХранилищаСертификатовКриптографии.Персональ
ныеСертификаты);
    Список = ПолучитьСписокСертификатов(75, ТипыСертификатов, Истина);
    // В примере всегда шифруем с помощью первого по порядку сертификата
    Сертификаты = Новый Массив;
    Сертификаты.Добавить(Список[0].Выгрузить());
    // Шифруем файл
    Результат = ЗашифроватьНаСервере(Адрес, Сертификаты);
    Если Не Результат Тогда
        Возврат;
    КонецЕсли;
    // Интерактивно сохраняем зашифрованный файл на диск
    ПолучитьФайл(Адрес, , Истина);
КонецПроцедуры
```

Листинг 3. Шифрование файла

Далее выполняется попытка расшифровки выбранного файла (листинг 4). Необходима реализация функции ПолучитьПарольДоступа(), которая возвращает значение пароля доступа к закрытому ключу. При использовании метода Расшифровать() исключение будет вызвано только после того, как неудачно завершились попытки расшифровки с помощью всех доступных сертификатов, а не после первой ошибки.

```
&НаКлиенте
Процедура РасшифровкаФайла()
    // Выбрать зашифрованный файл
    Адрес = "";
    Результат = ПоместитьФайл(Адрес, , , Истина);
    Если Не Результат Тогда
        Возврат;
    КонецЕсли;
    Данные = ПолучитьИзВременногоХранилища(Адрес);
    // Расшировать файл
    МенеджерКриптографии = Новый МенеджерКриптографии("", "", 75);
```

```
    МенеджерКриптографии.ПарольДоступаКЗакрытомуКлючу =
ПолучитьПарольДоступа();
    РасшифрованныеДанные = МенеджерКриптографии.Расшифровать(Данные);
    Адрес = ПоместитьВоВременноеХранилище(РасшифрованныеДанные);
    ПолучитьФайл(Адрес, , Истина);
КонецПроцедуры
```

Листинг 4. Расшифровка файла

Затем выполняется формирование файла с электронной цифровой подписью (листинг 5). В данном примере ЭЦП всегда сохраняется в файл signature.sign. Необходима реализация функции ПолучитьПарольДоступа(), которая возвращает значение пароля доступа к закрытому ключу.

```
&НаКлиенте
Процедура ПодписатьФайл()
    Адрес = "";
    Результат = ПоместитьФайл(Адрес, , , Истина);
    Если Не Результат Тогда
        Возврат;
    КонецЕсли;
    ТипыСертификатов = Новый Массив;
ТипыСертификатов.Добавить(ТипХранилищаСертификатовКриптографии.Персональ
ныеСертификаты);
    Список = ПолучитьСписокСертификатов(75, ТипыСертификатов, Истина);
    Сертификат = Список[0];
    Данные = ПолучитьИзВременногоХранилища(Адрес);
    МенеджерКриптографии = Новый МенеджерКриптографии("", "", 75);
    МенеджерКриптографии.ПарольДоступаКЗакрытомуКлючу =
ПолучитьПарольДоступа();
    МенеджерКриптографии.Подписать(Данные, "signature.sign",
Сертификат);
КонецПроцедуры
```

Листинг 5. Формирование ЭЦП

В заключении функция выполняет проверку (листинг 6) действительности ЭЦП для пары оригинальный файл – файл ЭЦП.

```
&НаКлиенте
Функция ПроверитьПодписьФайла(ИмяПодписанногоФайла, ИмяФайлаПодписи)
    Сертификат = Неопределено;
    МенеджерКриптографии = Новый МенеджерКриптографии("", "", 75);
    Попытка
        МенеджерКриптографии.ПроверитьПодпись(ИмяПодписанногоФайла,
ИмяФайлаПодписи, Сертификат);
    Исключение
        Возврат Ложь;
    КонецПопытки
    Возврат Истина;
КонецФункции
```

Листинг 6. Выполнение проверки

Использование асинхронной схемы работы с криптографией имеет ряд особенностей:

1. Она доступна только на стороне клиентского приложения. На стороне сервера следует использовать синхронный способ работы.
2. Инициализация объектов доступа к криптографической функциональности отличается от своих синхронных аналогов.

Рассмотрим пример работы с криптографической функциональностью в асинхронном режиме. Здесь будет рассмотрен процесс получения списка сертификатов (листинг 7) из определенного типа хранилища сертификатов.

```
&НаКлиенте
Процедура ПолучитьСписокСертификатов()
    ДополнительныеПараметры = Новый Структура;
    ДополнительныеПараметры.Вставить("ТипХранилищаСертификатов",
ТипХранилищаСертификатовКриптографии.ПерсональныеСертификаты);
    ДополнительныеПараметры.Вставить("ПроверятьДатуОкончания", Истина);
    ОписаниеОповещения = Новый
ОписаниеОповещения("ПолучитьСписокСертификатовЗавершение", ЭтотОбъект,
ДополнительныеПараметры);
    ДополнительныеПараметры.Вставить("ОповещениеЗавершения",
ОписаниеОповещения);
    ОбратныйВызов = Новый ОписаниеОповещения("ИнициализацияЗавершение",
ЭтотОбъект, ДополнительныеПараметры);
    Криптография = Новый МенеджерКриптографии;
    Криптография.НачатьИнициализацию(ОбратныйВызов, "", "", 75);
КонецПроцедуры
&НаКлиенте
Процедура ПолучитьСписокСертификатовЗавершение(СписокСертификатов,
ДополнительныеПараметры) Экспорт
    // здесь используем полученные сертификаты
КонецПроцедуры
&НаКлиенте
Процедура ИнициализацияЗавершение(Менеджер, ДополнительныеПараметры)
Экспорт
    Оповещение = Новый
ОписаниеОповещения("ПолучениеХранилищаСертификатовЗавершение",
ЭтотОбъект, ДополнительныеПараметры);
    Менеджер.НачатьПолучениеХранилищаСертификатов(Оповещение,
ДополнительныеПараметры.ТипХранилищаСертификатов);
КонецПроцедуры
&НаКлиенте
Процедура ПолучениеХранилищаСертификатовЗавершение(Хранилище,
ДополнительныеПараметры) Экспорт
    Оповещение = Новый
ОписаниеОповещения("МассивСертификатовЗавершение", ЭтотОбъект,
ДополнительныеПараметры);
    Хранилище.НачатьПолучениеВсех(Оповещение);
КонецПроцедуры
&НаКлиенте
Процедура МассивСертификатовЗавершение(Сертификаты,
ДополнительныеПараметры) Экспорт
    Результат = Новый Массив;
    ТекущаяДата = ТекущаяДата();
    Для Каждого Сертификат Из Сертификаты Цикл
        Если ДополнительныеПараметры.ПроверятьДатуОкончания И
Сертификат.ДатаОкончания < ТекущаяДата Тогда
            Продолжить;
        КонецЕсли;
        Результат.Добавить(Сертификат);
    КонецЦикла;

ВыполнитьОбработкуОповещения(ДополнительныеПараметры.ОповещениеЗавершени
я, Результат);
КонецПроцедуры
```

Листинг 7. Получение списка сертификатов в асинхронном режиме

Данный пример полностью иллюстрирует особенности работы с криптографией в асинхронном режиме. Получение списка сертификатов начинается с вызова метода ПолучитьСписокСертификатов() и заканчивается вызовом метода ПолучитьСписокСертификатовЗавершение(). Собственно получение состоит из инициализации модуля работы с криптографией и использования асинхронных аналогов следующих синхронных методов: ПолучитьХранилищеСертификатов() и ПолучитьВсе(). Также отдельным образом выполняется проверка на истекшие сертификаты. Оповещение ИнициализацияЗавершение() будет вызвано после того, как завершится инициализация менеджера криптографии. Затем работа строится аналогичным образом: в обработчике завершения предыдущего метода вызывается следующее действие, в обработчике завершения которого происходит очередное действие и т. д.

Получается четкая последовательность действий:
ПолучитьСписокСертификатов() → ИнициализацияЗавершение() →
ПолучениеХранилищаСертификатовЗавершение() → МассивСертификатовЗавершение() →
ПолучитьСписокСертификатовЗавершение()

Особого внимания требует только последний переход. В обработчике оповещения МассивСертификатовЗавершение() не вызывается никакого асинхронного метода, который требует указания обработчика оповещения. Нужное описание оповещения поступает в самый последний (по цепочке вызовов) обработчик оповещения с помощью задания дополнительных параметров. Нужные действия выполняются в методе ПолучитьСписокСертификатов(), а в самом последнем обработчике (ПолучитьСписокСертификатовЗавершение()) происходит явный вызов обработчика, «указатель» на который последовательно передавался по цепочки обработчиков с помощью дополнительных параметров. Фактический возврат происходит в результате выполнения метода ВыполнитьОбработкуОповещения().

Пример разработан для использования в модуле формы, но переработка примера для работы из общих модулей не представляет существенной проблемы. В этом случае первые две процедуры примера будут в одном общем модуле, а остальные – в другом, с соответствующей корректировкой описания обработчиков оповещения и вызовов методов. Аналогичным образом будут выглядеть и другие типичные приемы работы с криптографией.

Для работы с механизмом криптографии в веб-клиенте следует установить расширение работы с криптографией, при этом следует помнить, что во всех веб-браузерах, кроме Google Chrome, можно использовать как синхронные, так и асинхронные методы работы с криптографией. При этом используемый набор методов определяется свойством конфигурации «Режим использования синхронных вызовов расширений и внешних компонент». При использовании веб-браузера Google Chrome доступны только асинхронные методы работы с криптографией.

Список литературы

1. 1С:Предприятие 8. Система программ. URL: https:// https://v8.1c.ru (дата обращения: 14.06.2020).
2. Информационно-аналитический центр по автоматизации учета и управления. URL: https://infostart.ru (дата обращения: 15.07.2020).
3. Информационная система 1С:ИТС. URL: https://its.1c.ru/ (дата обращения: 01.09.2020).

EXAMPLES OF IMPLEMENTATION OF THE MECHANISM OF CRYPTOGRAPHY IN THE SYSTEM 1C: ENTERPRISE 8

S. V. Mishina
Senior Lecturer
svmishina2017o@mail.ru
Yelets

Bunin Yelets State University

Abstract. For a modern enterprise, the speed of business processes is one of the competitive advantages. When using electronic document management, time costs are reduced by at least 75%. In addition, there is a saving of own resources (paper, cartridges for printers, space for storing documents, employees' time for organizing paperwork). All this greatly helps the company. Almost every enterprise now uses remote banking channels, and the security of remote payments is currently very relevant. All major taxpayers now include in the contract theses that if you want to work with them, exchange documents electronically. All organizations are gradually switching to electronic invoices, electronic orders, etc. And if you want to participate in the auction, you must have an electronic signature. All this is gradually becoming widespread. But, despite all the ease of using these capabilities, there are factors that negatively affect the development of electronic document management. This is due to the complexity of protecting the data involved in this process. In this case, two goals are pursued: the confidentiality of personal and business information is ensured, and the accuracy of the stored information is guaranteed. For their implementation in the 1C: Enterprise 8 system, a cryptography mechanism is implemented. This article is devoted to a description of how to use it.

Keywords: encryption, cryptography, digital signature, public key, private key, certificate.

References

1. 1C:Enterprise 8. System programs. URL: https://its.1c.ru/ (date of access: 14.06.2020).
2. Information and analytical center for automation of accounting and management. URL: https://infostart.ru (date of access: 15.07.2020).
3. Information system 1C: ITS. URL: https://its.1c.ru/ (date of access: 01.09.2020).