Модуль 4. Алгоритмы и структуры данных

*Тема 4.15. Электронно-цифровая подпись

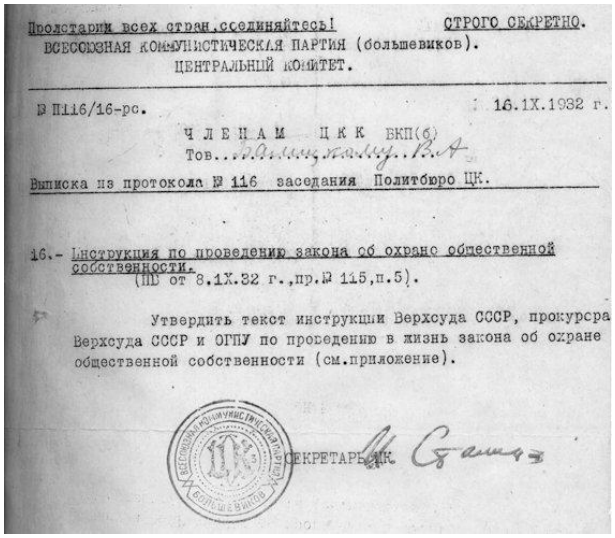
Оглавление

4.15. Электронно-цифровая подпись

Получали ли вы электронные сообщения в виде sms с незнакомых номеров, электронных писем с неизвестных адресов и т.п., которые были бы подписаны «как- будто» известными вам отправителями? Представляется, что такое происходило со всеми хотя бы единожды. Обычно в такой ситуации в голове крутятся мысли из разряда «Действительно ли я являюсь адресатом?» или «Действительно ли сообщение именно такого содержание могло прийти от этого адресанта?». Еще один пример. Клиент банка намерен перевести деньги со своего счета на счет какой-либо организации. При этом не вся передаваемая информация является конфиденциальной. Действительно, необходимо переслать лишь банковские реквизиты, которые общеизвестны и общедоступны. Однако банку важно убедиться, что деньги хочет перевести именно их владелец, а не злоумышленник. Клиент же заинтересован в том, чтобы сумма не была изменена и чтобы никто не смог ни переслать деньги от его имени и изменить информацию о получателе денег.

Во всех рассмотренных примерах информационная составляющая сообщений не является тайной. Вопрос лишь в её истинности, а так же в подлинности отправителя сообщений.

Действительно приказ считается вступившем в силу после его подписания соответствующим должностным лицом, счет может быть оплачен после визирования его главным бухгалтером и руководителем организации. Таким образом подпись является одним из главных реквизитов документа. Большинство официальных печатных/рукописных документов еще до недавнего времени визировались печатью и/или рукописной подписью.



Пролетарии всех стран, соединяйтесь! СТРОГО СЕКРЕТНО.
ВСЕСОЮЗНАЯ КОММУНИСТИЧЕСКАЯ ПАРТИЯ (большевиков).
ЦЕНТРАЛЬНЫЙ КОМИТЕТ.

№ П116/16-рс. 16.IX.1932 г.

ЧЛЕНАМ ЦКК ВКП(б)
Тов. [illegible]

Выписка из протокола № 116 заседания Политбюро ЦК.

16.- Инструкция по проведению закона об охране общественной собственности.
(ПБ от 8.IX.32 г., пр.№ 115,п.5).

Утвердить текст инструкции Верхсуда СССР, прокурора Верхсуда СССР и ОГПУ по проведению в жизнь закона об охране общественной собственности (см.приложение).

СЕКРЕТАРЬ ЦК

ООО «Дубрава»
115093, Москва, ул. Тверская, д. 7
Телефон: +7 (495) 123-77-77

Дубрава

Образец заполнения платежного поручения

ИНН 7705667798 КПП 775001001		
Получатель ООО «Дубрава»	Сч. №	40702810200642000265
Банк получателя ООО «УРАЛСИБ»	БИК	044525787
	Сч. №	30101810100000000787

Счет № 114 от 18 ноября 2010 года

Плательщик ПБЮЛ Вавилов Ю.Ю.
Получатель ООО «Дубрава»

№	Наименование товара	Количество	Цена	Сумма
1	Банщик ласковый	1	5 000,00	5 000,00
2	Веники дубовые ласковые	4	120,00	480,00
3	Шапки войлочные летние	4	280,00	1 120,00
4	Кадка деревянная древняя	1	800,00	800,00
			Итого	7 400,00
			Без налога (НДС)	-
			Всего к оплате	**7 400,00**

Всего четыре наименования, на сумму 7 400,00 р.
Семь тысяч четыреста рублей 00 копеек

Генеральный директор Платонов П. А.
Главный бухгалтер Платонов П. А.
Дубрава

В результате проникновения компьютерных технологий практически во все сферы деятельности человека возникла потребность реализовать аналог собственноручной подписи человека в электронном виде. Эта задача была успешно решена. В основе её решения лежат разработанные в середине 70-х годов криптографические алгоритмы с открытым ключом. На основе одного из них была реализована электронно-цифровая подпись (ЭЦП).

Однако, если собственноручная подпись конкретного человека всегда выглядит одинаково, то ЭЦП разная для различных документов, которые подписывает один и тот же автор.

Рассмотрим задачу. Пусть необходимо передать по открытому каналу связи электронный документ (причем, возможно, в незакодированном виде) и предоставить возможность получателю удостовериться, что дукумент передан в неискаженном виде и в том, что оправителем этого документа является лицо подписавшее его. Подобные задачи решаются с помощью ЭЦП. Как уже говорили ранее, ЭЦП это байтовая последовательность, прикрепеленная к заверяемому документу, формируемая специальным алгоритмом (Digital Signature Algorithm, DSA) на основе исходного текста электронного докуента и закрытого ключа. Помимо ЭЦП с передаваемым документом передается и так называемый открытый ключ.

Важно понимать, что открытый и закрытый ключи генерируются в паре. Для простоты воспиятия будем считать, что одному открытому ключу соответствует единственный закрытый и наоборот.

Тот же алгоритм (DSA) позволяет, получив на вход сам электронный документ, его ЭЦП и открытый ключ, ответить на вопрос действительно ли данная ЭЦП получена с помощью соответствующего закрытого ключа и именно из этого документа.

Установлением соответствий между парами ключей (открытым и закрытым) и их владельцами занимаются специальные организации - удостоверяющие центры. Удостоверяющий центр – это юридическое лицо, согласно Закону «Об электронно-цифровой подписи» выполняющее следующие функции:

- изготовление сертификатов ключей подписей;
- создание (генерация) ключей электронных цифровых подписей по обращению клиентов с гарантией сохранения в тайне закрытого ключа ЭЦП;
- приостановка и возобновление действия сертификатов ключей подписей, а также их аннулирование;
- ведение реестра сертификатов ключей подписей, обеспечение его актуальности и возможности свободного доступа к нему клиентов;
- проверка уникальности открытых ключей электронных цифровых подписей в реестре сертификатов ключей подписей и архиве удостоверяющего центра;
- выдача сертификатов ключей подписей в форме документов на бумажных носителях и (или) в форме электронных документов с информацией об их действии;
- осуществление по обращениям пользователей сертификатов ключей подписей подтверждение подлинности электронной цифровой подписи в электронном документе в отношении выданных им сертификатов ключей подписей;
- предоставление клиентам иных связанных с использованием электронных цифровых подписей услуги.

Резюмируя все вышенаписанное, можно определить свойства электронной подписи:

- Контроль целостности передаваемого документа: при любом случайном или преднамеренном изменении документа подпись станет недействительной, потому что вычислена она на основании исходного состояния документа и соответствует лишь ему.

- Защиту от изменений (подделки) документа: гарантия выявления подделки при контроле целостности делает подделывание нецелесообразным в большинстве случаев.
- Невозможность отказа от авторства. Так как создать корректную подпись можно, лишь зная закрытый ключ, а он должен быть известен только владельцу, то владелец не может отказаться от своей подписи под документом.
- Доказательное подтверждение авторства документа: Так как создать корректную подпись можно, лишь зная закрытый ключ, а он должен быть известен только владельцу, то владелец пары ключей может доказать своё авторство подписи под документом. В зависимости от деталей определения документа могут быть подписаны такие поля, как «автор», «внесённые изменения», «метка времени».

4.15.1 Основные понятия и термины

Digital Signature Algorithm (DSA) — криптографический алгоритм (который описывается в стандорте Digital Signature Standard) с использованием открытого ключа для создания электронной подписи. В отличие от RSA, который используется для шифрования информации, этот алгоритм испоьзуется для формирования ЭЦП. DSA используется одной стороной для генерации подписи данных, а другой - для проверки подлинности подписчика. Подпись генерируется при помощи закрытого ключа. Любая сторона может проверить подлинность цифровой подписи при помощи открытого ключа. Открытый и закрытый ключи не совпадают.

Закрытый ключ - байтовая последовательность “известная” только владельцу ЭЦП, с помощью которой происходит процедура подписания документа (генерация ЭЦП).

Открытый ключ - байтовая последовательность, которая известна лицам производящим верификацию подписанного документа (проверка ЭЦП).

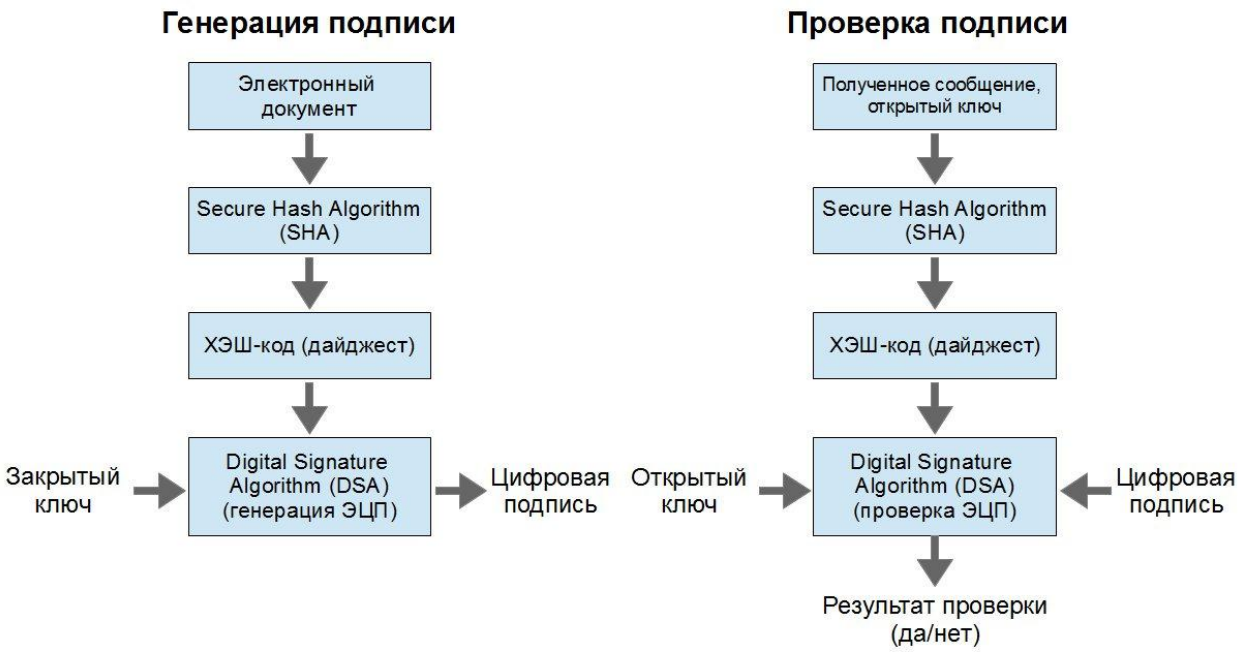
Асимметричные схемы ЭЦП относятся к криптосистемам с открытым ключом. В отличие от алгоритмов шифрования, в которых зашифрование производится с помощью открытого ключа, а расшифрование — с помощью закрытого (RSA является примером такого алгоритма), в схемах ЭЦП подписывание (именно подписывание - текст документа не шифруется!) производится с применением закрытого ключа, а проверка — с применением открытого.

Secure Hash Algorithm (SHA) - хэш-функция используемая при генерации подписи для получения сжатой версии документа (дайджеста).

4.15.2 Процедура создания и использования ЭЦП

Общепризнанная схема цифровой подписи охватывает три процесса:

1. Генерация ключевой пары.
2. Формирование подписи.
3. Проверка (верификация) подписи.



Алгоритм DSA. Генерация пары (открытого и закрытого) ключей

1. p – простое число p, где $2^{L-1} < p < 2^{L}$, 512 =< L =< 1024 и L кратно 64
2. q – простой делитель p-1, причем $2^{159} < q < 2^{160}$
3. $g = h^{(p-1)/q} \bmod p$, где h любое целое число $1 < h < p-1$ такое, что $h^{(p-1)/q} \bmod p > 1$
4. x – случайное или псевдослучайное целое число, где $0 < x < q$
5. $y = g^{x} \bmod p$
6. k – случайное или псевдослучайное целое число, где $0 < k < q$.

Целые p, q и g могут быть открытыми и могут быть общими для группы людей. x и y являются закрытым и открытым ключами, соответственно. Параметры x и k используются только для генерации подписи и должны держаться в секрете. Параметр k разный для каждой подписи.

Алгоритм DSA. Подпись сообщения с помощью закрытого ключа и параметра k

Подписью (ЭЦП) сообщения M является пара чисел r и s, где

1. $r = (g^{k} \bmod p) \bmod q$
2. $s = (k^{-1}(SHA(M) + xr)) \bmod q$.

SHA(M) — 160-битная бинарная строка, возвращаемая хэш-функцией.

Если r = 0 или s = 0, должно быть сгенерировано новое k и вычислена новая подпись. Если подпись вычислялась правильно, вероятность того, что r = 0 или s = 0 очень мала. Подпись вместе с сообщением пересылается получателю.

Алгоритм DSA. Проверка подписи

Числа p, q, g и открытый ключ находятся в открытом доступе.

Пусть M', r' и s' полученные версии M, r и s, соответственно, и пусть y — открытый ключ. При проверке подписи сначала нужно посмотреть, выполняются ли следующие неравенства: $0 < r' < q$ и $0 < s' < q$.

Если хотя бы одно неравенство не выполнено, подпись должна быть отвергнута. Если условия неравенств выполнены, производятся следующие вычисления:

$w = (s')^{-1} \bmod q$

$u1 = ((SHA(M')w) \bmod q$

$u2 = ((r')w) \bmod q$

$v = (((g)^{ul} (y)^{u2}) \bmod p) \bmod q.$

Если v = r', то подлинность подписи подтверждена.

Если v ≠ r', то сообщение могло быть изменено, сообщение могло быть неправильно подписано или сообщение могло быть подписано мошенником. В этом случае полученные данные следует рассматривать как поврежденные.

4.15.3. Создание и использования ЭЦП средствами языка Java.

Полностью программа будет иметь такой вид:

```
import java.net.ProxySelector;
import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.MessageDigest;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.SecureRandom;
import java.security.Signature;
import java.util.Scanner;


public class DSA {

        static PrivateKey privkey;
        static PublicKey pubkey;
        static Signature sigalg;

        public static void makeCrypto()
        {
        try{
        KeyPairGenerator keygen  =  KeyPairGenerator.getInstance("DSA");
        keygen.initialize(512, new SecureRandom()); //Для  генерации  ключей
нужно  инициализировать  объект-алгоритм,  указав  мощность  (strength)  ключа  и
безопасный  генератор случайных чисел
        KeyPair keys = keygen.generateKeyPair();
        pubkey = keys.getPublic ();
        privkey = keys.getPrivate ();
        sigalg = Signature.getInstance("DSA");   //Чтобы  подписать  сообщение
создается  объект  Signature,  реализующий алгоритм подписи.
        }
        catch(Exception ex)
        {
        }
        }


        public static byte[] makeDigest(String kod)
        {
```

```
        byte [] hash = null;
        try{
        MessageDigest alg = MessageDigest.getInstance ("SHA-1");
            alg.update(kod.getBytes());
        hash = alg.digest();
        }
        catch(Exception ex){
        }
        return hash;
        }


        public static byte[] makeSign(byte[] kod)
        {
        byte [] signature = null;
        try
        {
        sigalg.initSign(privkey);
        sigalg.update(kod);
        signature = sigalg.sign();
        }
        catch(Exception ex){

        };
        return signature;
  };

        public static void main(String[] args) {

                String message;
                String message2;
                String ecp2;
                Scanner in = new Scanner(System.in);

                makeCrypto();
                System.out.println("Открытый ключ: " + pubkey.toString());
                System.out.println("Закрытый ключ: " + privkey.toString());

                System.out.print("Введите текст исходного сообщения: ");
                message = in.nextLine();

                byte [] hcode = makeDigest(message);
                System.out.println("Дайджест (хэш-код) введенного сообщения: " +
hcode.toString());

                byte [] ecp = makeSign(hcode);
                System.out.println("ЭЦП введенного сообщения: " + ecp.toString());

                System.out.println("Передаваемая получателю информация: ");
                System.out.println("   1) Открытый ключ;");
                System.out.print("   2) Введите исходное сообщение: ");
                message2 = in.nextLine();
                System.out.print("   3) Введите ЭЦП: ");
                ecp2 = in.nextLine();

                try{
                sigalg.initVerify(pubkey);
                sigalg.update(makeDigest(message2));
```

```
        boolean verifies = sigalg.verify(ecp2.getBytes());
                if(verifies)
                System.out.println("Да");
                else
                System.out.println("Нет");
                }catch(Exception ex){

                }
        }

}
```

Упражнение 4.15.1

Доработать программу, чтобы сообщение подпись осуществлялась не ко всему сообщению, а к его дайджесту. Создать программу для проверки подписи сообщения. Вводятся сообщение, его подпись и открытый ключ. Необходимо вывести верно ли подписано сообщение.

Благодарности

Компания Samsung Electronics выражает благодарность за участие в подготовке данного материала преподавателю IT ШКОЛЫ SAMSUNG Хальзову Кирилу Сергеевичу.