Электронная цифровая подпись

К лекции
«Электронные деньги»
Ливак Е.Н.

ЭЦП – раздел криптографии

ЭЦП используется

1) для аутентификации автора (создателя) информации;
2) для доказательства (проверки) целостности,

т.е. того факта, что подписанное сообщение или данные не были модифицированы при передаче информации в компьютерных сетях.

Электронная цифровая подпись

- Битовая строка, присоединяемая к документу после подписания,

  называется электронной цифровой подписью.

- Протокол, с помощью которого получатель убеждается в подлинности отправителя и целостности сообщения, называется проверкой подлинности.

Построение ЭЦП

ЭЦП строится на основе двух компонент

- содержания информации, которая подписывается,
- и личной информации (код, пароль, ключ) того, кто подписывает.

Очевидно, что изменение каждой компоненты приводит к изменению электронной цифровой подписи.

Используемые алгоритмы

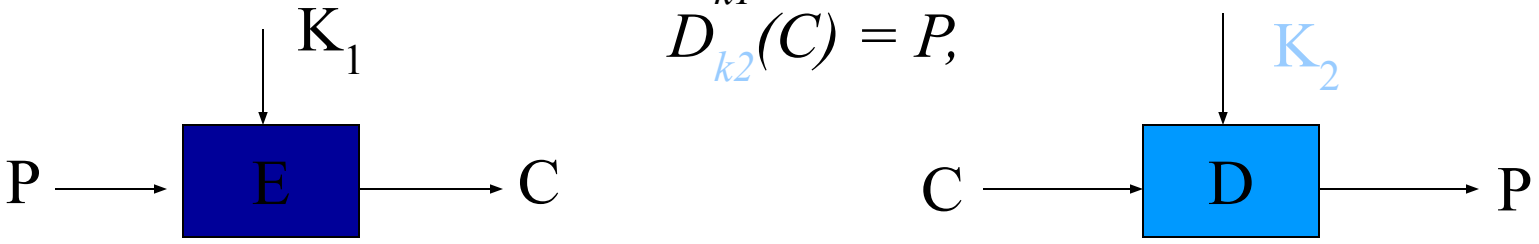
- Первые варианты цифровой подписи были реализованы с помощью симметричных криптосистем (специальные режимы функционирования).
- Современные процедуры создания и проверки ЭЦП основаны на шифровании с открытым ключом.

Напоминание

В асимметричных криптосистемах (системах с открытым ключом) используется:

1) **открытый** ключ – для шифрования
2) *соответствующий* ему **секретный** – для расшифрования:

$$E_{k1}(P) = C$$
$$D_{k2}(C) = P,$$



где k_1 – открытый ключ, k_2 – секретный ключ.

Применение криптографии с открытым ключом для создания ЭЦП

- основано на шифровании сообщения секретным (закрытым) ключом,
- и расшифровании – открытым.

$$E_{k1}(P) = C$$

$$D_{k2}(C) = P$$

где k_1 – секретный, k_2 – открытый ключ.

Протокол подписания документа

1. Алиса шифрует документ своим закрытым ключом (подписывает его)
2. Алиса посылает Бобу подписанный документ
3. Боб расшифровывает документ, используя открытый ключ Алисы (проверяя тем самым достоверность подписи)

Наиболее известные схемы создания ЭЦП

- DSA
- ECDSA (Elliptic Curve Digital Signature Algorithm)

алгоритм, аналогичный по своему строению DSA, но определённый не над кольцом целых чисел, а в группе точек эллиптической кривой.

- RSA
- Эль-Гамаля (ElGamal)
- Вероятностная схема подписи Рабина
- Схема Шнорра
- Диффи-Лампорта

Стойкость схем ЭЦП

- Стойкость распространенных схем ЭЦП основана на сложности решения частной задачи дискретного логарифмирования в простом поле *GF(*p*).*

- Задача эта формулируется следующим образом:
  - заданы простые числа *p, q* и натуральное число *g* < *p* порядка *q*, то есть

    $g^q \equiv 1 \ (mod\ p);$
  - зная значение $y = g^x \ (mod\ p)$,
  - необходимо найти $x \in Z$.

DSA – Digital Signature Algorithm алгоритм цифровых сигнатур

- Алгоритм с открытым ключом.
- Размер ключа – от 512 до 1024 бит.
- При проверке достоверности сигнатуры DSA работает в 10-40 раз медленнее RSA.
- Используется только для ЭЦП, не для шифрования.

- DSS – стандарт США (1994) на основе DSA и алгоритме хэширования SHA-1.

Алгоритм DSA.
Подпись сообщения

1. Алиса генерирует случайное число k, меньшее q
2. Алиса генерирует

   $r = (g^k \bmod p) \bmod q$

   $s = (k^{-1} (H(m) + x\, r)) \bmod q$

   H(m) – хэш-функция
3. Алиса посылает Бобу свою подпись - (r, s)

Алгоритм DSA.
Проверка подписи

1. Боб проверяет подпись
   - Вычисляет

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

     $u_1 = (H(m) * w) \bmod q$

     $u_2 = (r\ w) \bmod q$

     $v = ((g^{u1} * y^{u2}) \bmod p) \bmod q$

2. Если $v = r$, то подпись верна

Национальные стандарты (первые)
(реализовывали ЭЦП в простом поле)

- Стандарт США
  - **Digital Signature Standard** (принят в 1991 г. с последующими изменениями в 1993, 1996 г.)
- Российский стандарт цифровой подписи
  ГОСТ Р 34.10-94
  Разработан Главным управлением безопасности связи ФАПСИ
- Стандарт Республики Беларусь
  СТБ 1176.2 (1999 г.)

Стандарты Российской Федерации

- 1994 год - ГОСТ Р 34.10-94 «Информационная технология. Криптографическая защита информации. Процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма»
- 2002 год - ГОСТ Р 34.10-2001

  основан на вычислениях в группе точек эллиптической кривой

  В соответствии с этим стандартом, термины «электронная цифровая подпись» и «цифровая подпись» являются синонимами
- 01.01. 2013 - ГОСТ Р 34.10-2012 «Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи»

Белорусские стандарты ЭЦП

- СТБ 1176.2-99 (в настоящее время не действует)

- СТБ 34.101.45-2013

  Информационные технологии и безопасность АЛГОРИТМЫ ЭЛЕКТРОННОЙ ЦИФРОВОЙ ПОДПИСИ И ТРАНСПОРТА КЛЮЧА НА ОСНОВЕ ЭЛЛИПТИЧЕСКИХ КРИВЫХ

  Рекомендация преподавателя

  С криптографическими стандартами РБ лучше ознакомиться на страничке **НИИ прикладных проблем математики и информатики БГУ**

  http://apmi.bsu.by/resources/std.html

Недостатки существующих схем формирования ЭЦП

1) медленная работа алгоритмов формирования и проверки подписи;

2) ограничения на длину подписываемого сообщения.

Решение проблемы, связанной с ограничениями на длину

– разбиение сообщения на фрагменты и подпись каждого фрагмента

Однако ⇒

- увеличение объема сообщения
- и времени выполнения процедур создания и проверки ЭЦП

Механизм хэш-функций

Применяется

- для уменьшения времени, необходимого для генерации и проверки подписи,
- для сокращения длины ЭЦП

Обычно для использования алгоритма создания ЭЦП необходимо, чтобы подписываемое сообщение являлось числом. Хеш-функция должна преобразовать любое сообщение в последовательность битов, которые можно потом преобразовать в число.

«Основная работа любой хэш-функции заключается в превращении (или хэшировании) произвольного набора элементов данных в значение фиксированной длины («отпечатка» или «дайджеста»)»

Механизм хэш-функций для ЭЦП

- подписанное сообщение *m* будет иметь вид

$$(m, S(h(m))),$$

где S – функция выработки подписи,

h - односторонняя хэш-функция

Хэш-функцией называется всякая функция

h: X -> Y,

легко вычислимая и такая, что для любого сообщения *M* значение

h(M) = H (свертка, хэш-код)

имеет фиксированную битовую длину.

X — множество всех сообщений,

Y — множество двоичных векторов фиксированной длины.

Построение функций хэширования

Требуется, чтобы вычислительно трудным являлось решение следующих криптоаналитических задач:

1) по заданному $y = h(x)$ определить x
(односторонняя функция h);

2) для заданного x найти другое x', такое, что $h(x) = h(x')$ (свободная от коллизий функция h);

3) 3) найти пару x, x′ (x ≠ x′), такую, что h(x) = h(x′)
(строго свободная от коллизий функция h).

Значение хэш-функции также называют

- Хэш-код
- Свертка
- Функция (значение) свертки
- Профиль сообщения
- Дайджест сообщения
- Криптографическая контрольная сумма
- Цифровой отпечаток
- Код аутентичности сообщения
- Код обнаружения манипуляций

Функции хэширования (Алгоритмы создания дайджестов сообщений)

Разработаны Рональдом Ривестом

- MD2 - Message Digest #2

Низкоскоростной, но очень надежный алгоритм, создающий 128-разрядные дайджесты данных любого объема.

- MD4 - Message Digest #4 (1990)

Более скоростной, но менее надежный алгоритм, создающий 128-разрядные дайджесты данных любого объема. 512-битовые блоки. Есть дефекты.

- MD5 Message Digest #5 (1992)

Версия MD4 с повышенной надежностью, преимущества также и в скорости. 128-разрядные дайджесты данных любого объема.

Не устойчив к коллизиям!!! ⇒ Не используется для долговременных ЭЦП

Функции хэширования

- **SHA - Secure Hash Algorithm** (1992)

  160-разрядные хэш-код (дайджест). НЕ устойчив к коллизиям.

  512-битовые блоки

- **SHA-1 - Secure Hash Algorithm 1** (1995)

  Модификация SHA. Исправлены недостатки. Решает проблему коллизий

- Хеш-функции **семейства SHA-2** (разработаны Агентством национальной безопасности США в 2002 г). Алгоритмически похожи на **SHA-1**

- **SHA-256 -** частный случай алгоритма из семейства SHA-2
  Используется в криптовалютах (майнинг), протоколах SSL, SSH, PGP и многих других.

- **CRIPT**

  Продвинутая и улучшенная версия алгоритма SHA-256 (разрабатывалась для усложнения аппаратных реализаций SHA-256 посредством увеличения количества ресурсов, которые необходимы для вычисления, а именно ОП)

  Используется в криптовалютах (майнинг)

- **SHA-3**

  В 2007 г. был объявлен конкурс на разработку функции, которая базируется на совершенно ином по сравнению с SHA-1 и SHA-2 алгоритме. 2 октября 2012 года NIST утвердил в качестве SHA-3 алгоритм Keccak

Функции хэширования

- Выделяют два важных вида криптографических хэш-функций — ключевые и бесключевые.

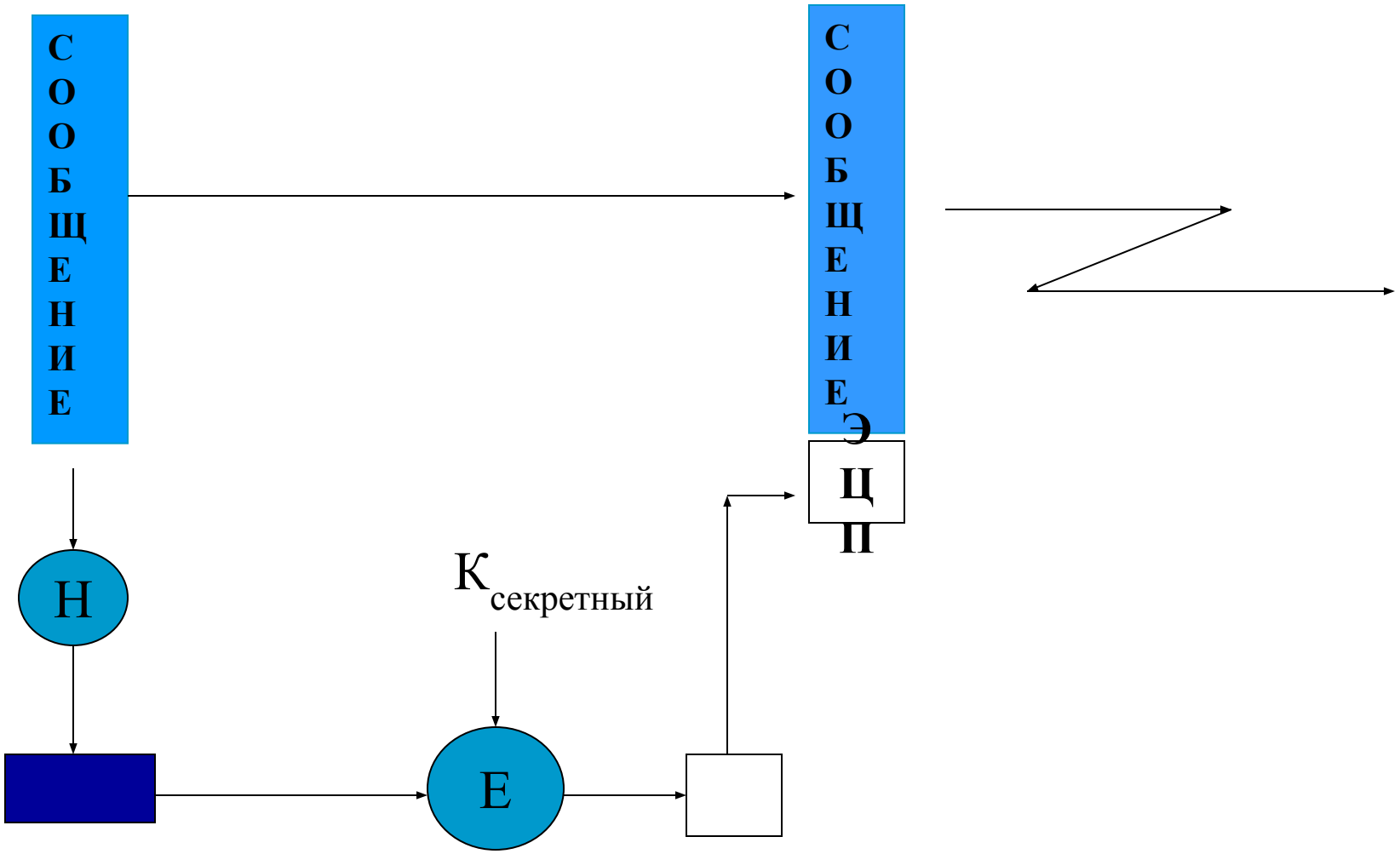
  Ключевые хэш-функции называют кодами аутентификации сообщений. Они дают возможность без дополнительных средств гарантировать как правильность источника данных, так и целостность данных в системах с доверяющими друг другу пользователями

Примеры ключевых функций хэширования

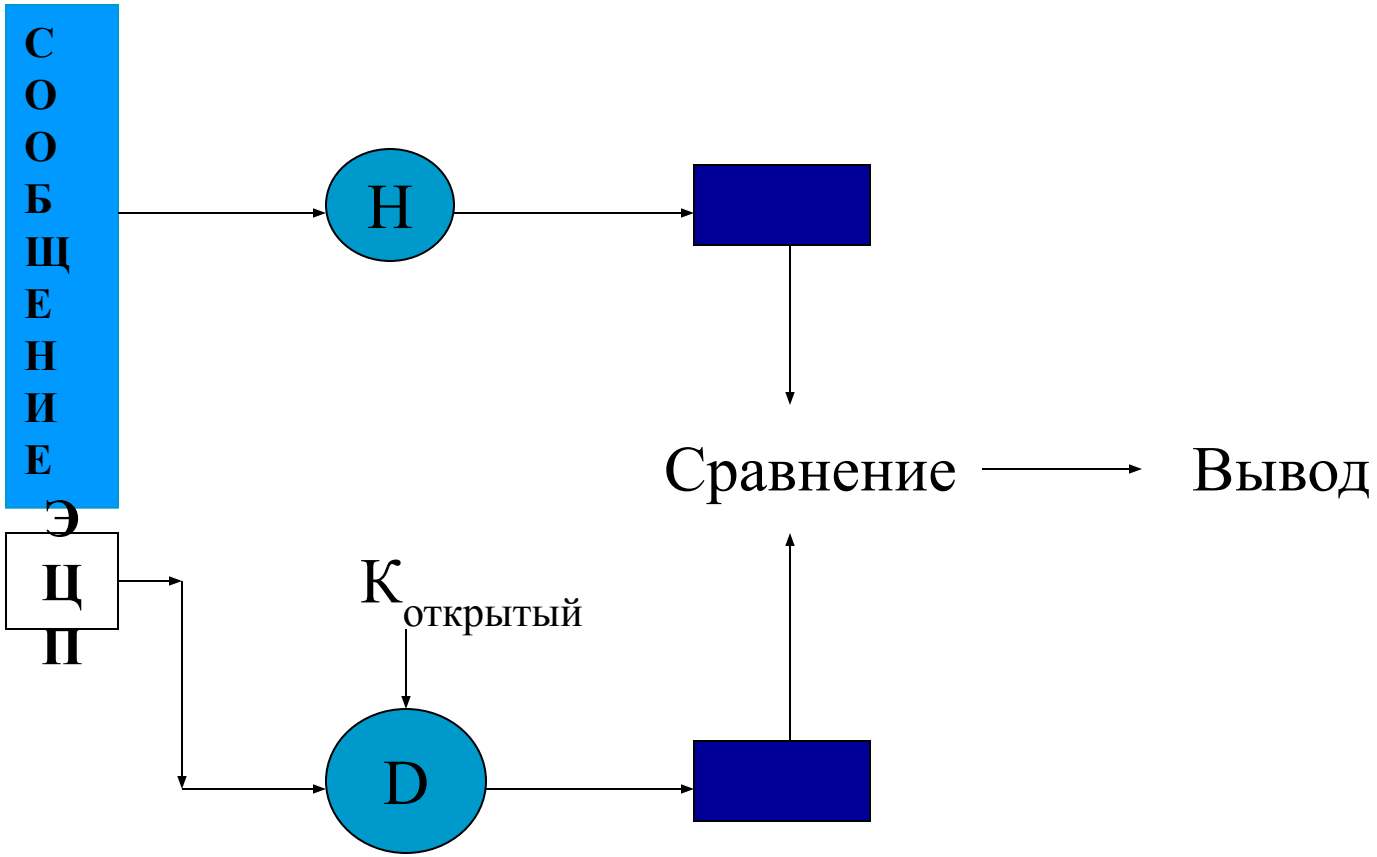
- **MAC - Message Authentication Code (код аутентификации сообщений)**
- **HMAC** При создании хэша (дайджеста) используется также секретный ключ. Использует 128-битную хэш-функцию

Бесключевые хэш-функции называются кодами обнаружения ошибок. Они дают возможность с помощью дополнительных средств (шифрования, например) гарантировать целостность данных. Эти хэш-функции могут применяться в системах как с доверяющими, так и не доверяющими друг другу пользователями.

Создание ЭЦП



Проверка целостности



Компрометация схем ЭЦП

- В настоящее время реализованы и опубликованы схемы механизмов взлома ЭЦП, основанные

  на генерации новой пары (открытый, секретный) ключей и включении нового открытого ключа в конверт ЭЦП

Механизм генерации и распределения ключей

распределение осуществляется двумя способами:

1) созданием центра генерации и распределения ключей;
2) прямым обменом ключами между абонентами.

- В первом случае компрометация центра приводит к компрометации всей передаваемой информации.
- Во втором случае – необходимо обеспечить подлинность каждого абонента.

Ненадежность практических реализаций алгоритмов ЭЦП

Анализ уязвимостей существующих схем ЭЦП позволяет утверждать, что

«число уязвимых точек ЭЦП, базирующейся на шифровании с открытым ключом, настолько велико, что целесообразность использования подобного метода вызывает большие сомнения»

[Ивт И., Богданов В. Надежна ли цифровая подпись?].