СПбГЭТУ «ЛЭТИ»
ПЕРВЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ

А.Б. Левина

Криптография и криптографические протоколы

Протоколы индивидуальной и коллективной цифровой

СПбГЭТУ «ЛЭТИ», 2022 г.

1. ВВЕДЕНИЕ

Мы продолжаем знакомство с протоколом аутентификации, на этой лекции мы познакомимся с протоколами мультиподписи, коллективной подписи и с применением симметричного шифрования для создания хэш-функций.

Сегодня на лекции будет рассмотрено:

- Понятие коллективной и мультиподписи;
- Атака «Дней Рождений»;
- Создание хэш-функций с помощью симметричного шифрования.

2. ЦИФРОВАЯ ПОДПИСЬ

2.1 СХЕМА ЦИФРОВОЙ ПОДПИСИ

Для начала вспомним, что такое цифровая подпись. В криптографии есть два вида схем цифровой подписи.

1. Схема подписи **с приложением**;
2. Схема подписи **с восстановлением сообщения**.

Схема подписи с приложением применяется в тех случаях, когда нас интересует только факт подписания документа, само сообщение, в данном случае, не интересует. Важна только информация, кто является отправителем.

Сообщение подписывается с помощью закрытого ключа, а проверяется подпись с помощью открытого ключа, получается, что проверить подпись может любой.

СООБЩЕНИЕ + секретный ключ Алисы = ПОДПИСЬ
СООБЩЕНИЕ + ПОДПИСЬ + ОТКРЫТЫЙ КЛЮЧ АЛИСЫ = ДА/НЕТ

Рис. 1: Схема подписи с приложением

Схема подписи с восстановлением не только предоставляет информацию, кому принадлежит сообщение, но и восстанавливает сообщение, в случае если подпись является ликвидной.

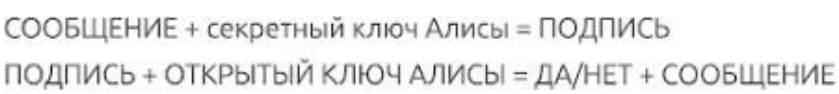

Рис. 2: Схема подписи с восстановлением

Цифровая подпись должна удовлетворять следующим критериям:

1. Подпись достоверна. Она убеждает получателя документа в том, что подписавший осознанно подписал документ.
2. Подпись неподдельна. Она доказывает, что именно подписавший, и никто иной, подписал документ.
3. Подпись не может быть использована повторно. Она является частью документа, ее невозможно перенести на другой документ.
4. Подписанный документ нельзя изменить.
5. От подписи нельзя отказаться. Подписавший не сможет утверждать, что он не подписывал документ, так как документ подписывается закрытым ключом, а пара закрытый ключ - открытый ключ уникальна.

Схематически это выглядит следующим образом:

- Алиса подписывает сообщение с помощью своего закрытого ключа, используя любой из алгоритмов постановки подписи. $S = s(M)$ -цифровая подпись.
- Боб осуществляет проверку подписи V (S), используя открытый ключ Алисы и получает на выходе сообщение M и бит v - результат проверки подписи.

Подпись гарантирует:

1. Целостность сообщения.
2. Оригинальность сообщения.

3. Отсутствие ренегатства=невозможность отказаться.

Мы должны помнить, что подпись ставится не на сообщение, а на его хэш.

Определение 2.1.1 *Хэш-функцией (хэш-кодом) называется преобразование, переводящие битовую строку любой длины в битовую строку фиксированной длины.*

Рассмотрим это на схеме:

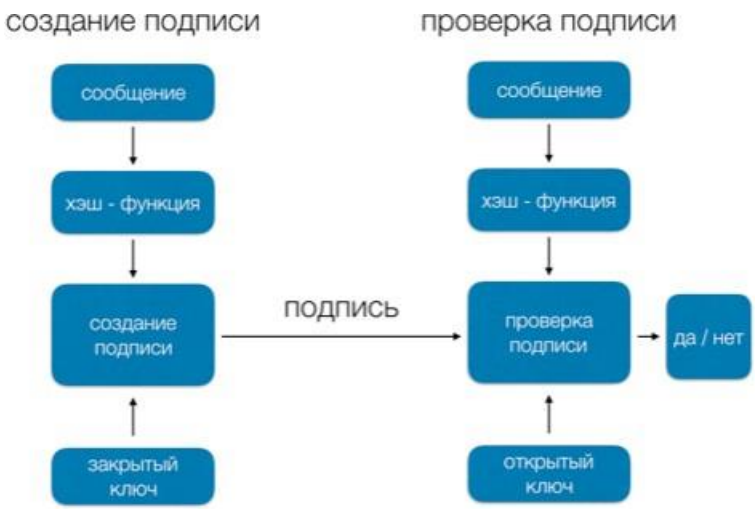

Рис. 3: Алгоритм постановки подписи

Давайте теперь рассмотрим процесс постановки нескольких подписей.

2.2 ПОСТАНОВКА НЕСКОЛЬКИХ ПОДПИСЕЙ/КОЛЛЕКТИВНОЙ ПОДПИСИ

***Определение 2.2.1 Множественная** (коллективная) **подпись** — схема (протокол) реализации электронной подписи (ЭЦП), которая позволяет нескольким пользователям подписывать единый документ.*

Коллективная подпись предоставляет возможность одновременного подписания электронного документа, поскольку формируется в результате единого неделимого преобразования и не может быть разделена на индивидуальные подписи; кроме этого, её нельзя расширить, то есть встроить в неё дополнительную подпись ещё одного или нескольких лиц

В ситуации, когда данным можно верить только если их подписало несколько объектов необходимо рассмотреть алгоритм постановки нескольких подписей. Сейчас нас не интересует вопрос какой именно алгоритм подписи использовался, а важно только шаги постановки подписей.

1. Алиса подписывает значение хэш-функции документа;
2. Боб подписывает значение хэш-функции документа;
3. Боб посылает свою подпись Алисе;
4. Алиса посылает Кэрол свою подпись, подпись Боба и подписанный документ;
5. Кэрол проверяет подпись Алисы и Боба.

После этих шагов Кэрол принимает решение может ли она доверять документу или нет.

Рассмотрим еще понятие мультиподпись.

Определение 2.2.2 ***Мультиподпись*** *— схема реализации электронной подписи, которая для своей достоверности требует* ***T*** *ключей из группы* ***N*** *членов (****T*** *<* ***N****). По своей сути является вариантом пороговой подписи, но реализована не как единый объект, а как проверка заданных условий, которую осуществляет базовая система скриптов криптовалюты.*

3. АТАКА «ДНЕЙ РОЖДЕНИЙ»

Рассмотрим одну из атак на мультиподпись, атака основана на нахождении коллизий в хэш-функции.

Из-за чего возникают коллизии? Интуитивно мы понимаем, что, сжимая строки сколь угодно большой длины в строки фиксированной длины, будут повторения. Математически наличие коллизий объясняется парадоксом дней рождения, познакомимся с ним.

3.1 ПАРАДОКС ДНЕЙ РОЖДЕНИЯ

Для начала рассмотри пример с шариками. Допустим, в ящике находятся m шариков разного цвета. Из ящика не глядя достают один шарик, записывают его цвет,

возвращают его в ящик и тянут снова. Вероятность того, что после n вытаскиваний нам попадется хотя бы два шарика одного цвета, равна:

$$1 - m^{(n)} / m^n$$

где $m^{(n)} = m \cdot (m-1) \cdot (m-2) \ldots (m-n+1)$.

Попытаемся понять причину названия парадокса. С помощью записанной формулы оценим вероятность совпадения дней рождений двух людей, находящихся в одной группе. Считается, что такая вероятность крайне мала, но, на самом деле, это не так.

Рассмотрим группу из 23 человек и не високосный год, соответственно дней в году 365. Подставляя эти значения в нашу формулу мы получаем, что вероятность совпадения двух дней рождений в один день равна:

$$1 - 365^{(23)} / 365^{23} \approx 0,5$$

Что является довольно большой вероятностью для такой маленькой группы. Как раз наличие этого парадокс и доказывает высокую вероятность происхождения коллизий.

Мультиподписные адреса Биткойна используют P2SH и защищены с помощью HASH160 (160-битная хеш-функция). Если злоумышленник владеет хотя бы 1 ключом из мультиподписного списка, то с учётом коллизии хеша для подделки чужой подписи он может снизить количество вариантов перебора до 2^{80}, что уже осуществимо для современных вычислительных систем.

4. БЛОЧНОЕ ШИФРОВАНИЕ КАК СПОСОБ СОЗДАНИЯ ХЭШ-ФУНКЦИЙ

Когда мы с Вами рассматривали структуру Меркла-Дамгарда, упоминалось, что функция сжатия может быть либо специально разработана для хеширования, либо представлять собой функцию блочного шифрования.

Рассмотрим как же можно применять алгоритмы блочного шифрования для создания хэш-функций.

Существует несколько наиболее известных структур. Все эти структуры используют схожие схемы. Сообщение дополняется до нужной длины и разбивается на группы x_1, $\ldots x_t$, длина которых совпадает или с размером блока или с длиной ключа, используемого блочным шифром. Выбор размера групп зависит от того, какой длины хэш-значение нам нужно. Значение конструируемой хэш-функции совпадает со значением H_t,

полученным итерацией следующих действий:

$$H_0 = IV,\ H_i = f\ (x_i,\ H_{i1}\ ),$$

где IV - инициализирующей вектор.

Рассмотрим наиболее известные структуры.

- **СТРУКТУРА ДЭВИСА-МЕЙЕРА**

Блок сообщения m_i и предыдущее значение хеш-функции H_{i-1} поступают в качестве ключа и блока открытого текста, соответственно, на вход блоного шифра E. Получившийся блок закрытого текста XOR-ся с предыдущим результатом H_{i-1} и такая итерация производится необходимое количество раз.

Математически это можно записать следующим образом:

$$H_i = E_{m_i}(H_{i-1}) \oplus H_{i-1}.$$

Как мы видим из этой схемы, в нашем алгоритме блочного шифрования, подключами являются блоки открытого текста, а открытым текстом – предыдущее хэш-значение. Длина блока m_i, равна длине подключа, необходимого для блочного алгоритма. Обратим внимание, что во всех схемах можно выбирать любой алгоритм блочного шифрования.

Работа схемы Дэвиса-Мейера представлена ниже.

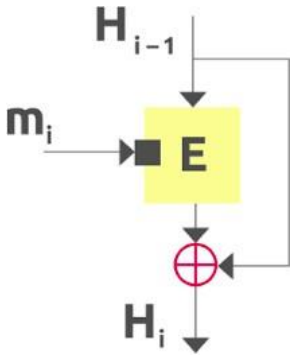

Рис. 4: Схема Дэвиса-Мейера

Следующая структура, с которой мы познакомимся - структура Матиса– Мейера – Осеаса.

• СТРУКТУРА МАТИСА — МЕЙЕРА — ОСЕАСА

Структура Матиса — Мейера — Осеаса является немного измененной версией схемы Девиса — Мейера. Данная схема может быть использована, если блоки данных и ключ шифрования имеют один и тот же размер, например как в алгоритм AES.

В данной схеме блок сообщения m_i и предыдущее значение хеш-функции H_{i-1} поступают в качестве ключа и блока открытого текста на вход блочного шифра E. Значение H_{i-1} подвергается предварительной обработке функцией g. Функция g реализует отображение n-битного значения хеш-функции в k-битный ключ шифра E. В результате применения операции шифрования, получается блок закрытого текста, который суммируется с соответствующим ему блоком открытого текста m_i.

Математически схема Матиса — Мейера — Осеаса записывается следующим образом:

$$H_i = E_{g(H_{i-1})}(m_i) \oplus m_i.$$

Как мы видим, подключом в этом случае является $g(H_{i-1})$, а открытым текстом - блоки m_i.

Схематично это выглядит следующим образом:

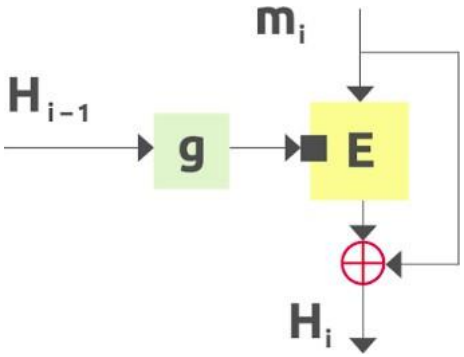

Рис. 5: Схема Матиса — Мейера — Осеаса

Последняя структура, с которой мы сегодня познакомимся, это схема Миагути — Пренеля.

• СТРУКТУРА МИАГУТИ — ПРЕНЕЛЯ

Данная схема является расширенной версией Матиса — Мейера — Осеаса. Здесь, блок закрытого текста XOR-ся не только с соответствующим ему блоком открытого текста m_i, но и с результатом предыдущей итерации хеширования H_{i-1}.

Математически это выглядит следующим образом:

$$Hi = E_{g(Hi-1)}(mi) \oplus Hi-1 \oplus mi$$

На рисунке 6 представлена работа структуры Миагути — Пренеля.

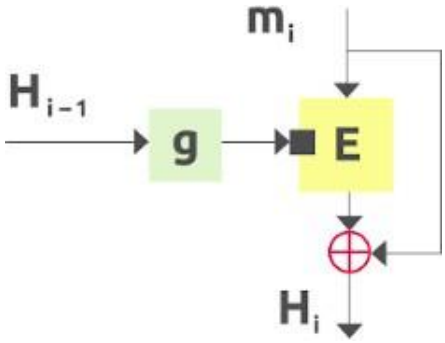

Рис. 6: Схема Миагути — Пренеля

Данная схема применяется в алгоритме Whirlpool. Это криптографическая хеш-функция, разработанная Винсентом Рэйменом и Пауло Баррето, которая хеширует входное сообщение с длиной до 2^{256} битов, а выходное значение составляет 512 битов.

Мы рассмотрели основные схемы использования блочных алгоритмов шифрования для создания хэш-функций, сама идея довольна проста и наглядна, но несмотря на это, алгоритмы хэширования, созданные на основе блочных шифров не получили широкого применения.

5. ЗАКЛЮЧЕНИЕ

На данной лекции мы познакомились с протоколом мультиподписи и коллективной подписи, рассмотрели атаку дней рождений, и использование симметричного шифрования для создания хэш-функции.