# **ЛР №11 Криптография. Исследование криптографических хеш-функций**

**Определение 1. Хеш-функция** – математическая или иная функция h = H(М), которая принимает на входе строку символов М, называемую также прообразом, переменной длины n и преобразует ее в выходную строку фиксированной (обычно – меньшей) длины l.

**Определение 2. Хеширование** – это преобразование входного массива данных определенного типа и произвольной длины (практически) в выходную битовую строку фиксированной длины.

Все существующие **функции хеширования можно разделить на два больших класса**:

• бесключевые хеш-функции, зависящие только от сообщения;

• хеш-функции с секретным ключом, зависящие как от сообщения, так и от секретного ключа.

**Определение 3. Криптографическая хеш-функция** – это специальный класс хеш-функций, который имеет различные свойства, необходимые для решения задач в области криптографии.

**Основные задачи**, решаемые с помощью хеш-функций:

• аутентификация (хранение паролей);

• проверка целостности данных;

• защита файлов;

• обнаружение зловредного ПО;

• криптовалютные технологии

**К основным свойствам хеш-функций** можно отнести следующие.

**Свойство 1. Детерминированность:** независимо от того, сколько раз вычисляется H(M), M – const, при использовании одинакового алгоритма код хеш-преобразования h всегда должен быть одинаковым.

**Свойство 2. Скорость вычисления** **хеша** h: если процесс вычисления h недостаточно быстрый, система просто не будет эффективной.

**Свойство 3. Сложность обратного вычисления**: для известного H(М) невозможно (практически) определить М. Это важнейшее свойство хеш-функции для криптографических применений – свойство односторонности преобразования

В плане односторонности хешей на основе блочных шифров отметим одно обстоятельство. Блочный шифр необратим по ключу шифрования. Основным недостатком хеш-функций на основе блочных шифров является сравнительно невысокая производительность.

**Свойство 4. Даже минимальные изменения** в хешируемых данных (М ≠ М') должны изменять хеш: Н(M) ≠ Н(М').

**Определение 4. Коллизией хеш-функции** Н называют ситуацию, при которой различным входам (в общем случае – х и у или М ≠ М') соответствует одинаковый хеш-код: H(x) = H(y) или H(М) == H(М').

**Свойство 5. Коллизионная устойчивость** (стойкость).

Зная М, трудно найти такое М' (М ≠ М'), для которого H(М) = H(М').

Если последнее равенство выполняется, то говорят о коллизии 1-го рода.

Если случайным образом выбраны два сообщения (М и М'), для которых H(М) = H(М'), говорят о коллизии 2-го рода.

*Мерой криптостойкости хеш-функции считается вычислительная сложность нахождения коллизии.*

Для того чтобы хеш-функция H(M) считалась криптографически стойкой, она должна удовлетворять трем основным требованиям: необратимостью вычислений (свойство 3), устойчивостью к коллизиям первого рода и устойчивостью к коллизиям второго рода (свойство 5).

**Определение 5. Хеш-функция** – это функция, выполняющая отображение из множества М в число, находящееся в интервале [0, m – 1]: h: M → [0, m – 1].

## **Парадокс «дней рождений» и его использование в криптографических приложениях**

Парадокс "дней рождений" говорит о том, что в группе из относительно небольшого числа людей существует высокая вероятность того, что двое из них будут иметь одинаковый день рождения. Это может показаться неожиданным, но объясняется большим количеством возможных комбинаций дней рождений.

Этот парадокс может быть использован и в криптографии. В частности, при атаке на хеш-функцию, где хеш-функция преобразует данные в короткий код, существует вероятность возникновения коллизии, то есть двух разных входных данных, дающих одинаковый хеш-код.

Вероятность коллизии в хеш-функции может быть оценена с использованием аналогии с парадоксом "дней рождений". Если хеш имеет длину l бит, то вероятность коллизии будет увеличиваться с ростом числа обрабатываемых данных. Важно выбирать достаточно длинный хеш для обеспечения криптостойкости и устойчивости к атакам.

Таблица, приведенная в тексте, показывает вероятностные оценки коллизий для различных длин хеш-функций. Чем больше длина хеша, тем меньше вероятность коллизии.

Важно учитывать стойкость хеш-функций к коллизиям при разработке криптографических протоколов и систем, чтобы обеспечить их безопасность и надежность.

## **Структурные и функциональные особенности некоторых хеш-функций**

Алгоритмы семейства MD-x (MD-6 не стандарт.)

MD 2,4,5 128 бит, MD6 от 1 до 512 бит

Алгоритмы семейства SHA являются в настоящее время широко распространенными. SHA-2 – собирательное название алгоритмов 224, 256, 384, 512.

SHA-3, одной из отличительных особенностей которого является использование конструкции криптографической губки». В этой конструкции реализован итеративный подход для создания функции с произвольной длиной на входе и произвольной длиной на выходе на основе определенного преобразования.

Максимальный объем хешируемых сообщений для алгоритмов SHA-1, SHA-256, SHA-224 такой же, как и для алгоритмов MD.

***Базовые алгоритмы обоих рассматриваемых семейств (MD и SHA) условно можно разделить на 5 стадий:***

• расширение входного сообщения;

• разбивка расширенного сообщения на блоки;

• инициализация начальных констант;

• обработка сообщения поблочно (основная процедура алгоритма хеширования);

• вывод результата.

Процедура расширения хешируемого сообщения достаточно подробно описана в [3] на примере алгоритма MD-4 (см. п. 10.3.1).

Входное сообщение «дополняется» (расширяется) так, чтобы его длина (в битах) была конгруэнтной к 448 по модулю 512. Это значит, что сообщение начальной длиной L битов расширяется так, что остаются незаполненными всего лишь 64 бита, чтобы итоговая длина L' была кратной 512. В указанные 64 бита записывается двоичная длина. Расширение происходит всегда, даже если длина сообщения уже соответствует 448, по модулю 512. Эта операция выполняется следующим образом: один бит «1» добавляется к сообщению, а затем добавляются биты «0», так что длина в битах дополненного сообщения стала конгруэнтной 448 по модулю 512. Добавляется не менее одного бита, но не более 448 битов.

Пусть исходное сообщение, или первообразная, будет словом «CRYPTO» (М = «CRYPTO»), или в кодах ASCII – это будут десятичные (67-82-89-80-84-79) и соответствующие двоичные (01000011-01010010-01011001-01010000-01010100-01001111;

здесь для отделения чисел используются дефисы) числа. Как видим, длина L хешируемого сообщения равна 48 битам. Эту длину мы должны расширить до 448 битов, добавив одну «1» и 399 «0». В последнюю часть из 64 битов полученного 512-битного модуля (L' = 512) мы запишем справа двоичное представление числа L = 48: 110000. В остальные 58 разрядов (из 64) мы впишем «0». После этого полученный расширенный блок делим на 16 32-разрядных подблоков.