Реализация криптографических алгоритмов в ИС

Лекция 2
Хэш-функции. Назначение и свойства хэш-функций (целостность, устойчивость к коллизиям).

Алгоритмы: SHA-256, SHA-3, ГОСТ Р 34.11-2012 ("Стрибог").

Содержание лекции

1. **Введение. Определение и назначение хэш-функций**
2. **Криптографические свойства хэш-функций**
3. **Обзор современных алгоритмов хэширования (алгоритмы: SHA-256, SHA-3, ГОСТ Р 34.11-2012 ("Стрибог")).**

Лекция 2
Хэш-функции. Назначение и свойства хэш-функций (целостность, устойчивость к коллизиям).
Алгоритмы: SHA-256, SHA-3, ГОСТ Р 34.11-2012 ("Стрибог").

1. Введение. Определение и назначение хэш-функций

1. Введение. Определение и назначение хэш-функций

Вычисление хэш-функций – один из основных и важнейших методов криптографической защиты информации, позволяющий с высокой вероятностью утверждать факт неизменности (целостности) данных при их передачи по сети и хранении в информационных системах.

Хэш-функции также применяются при использовании электронной подписи и шифровании данных.

1. Введение. Определение и назначение хэш-функций

Ключевое назначение хэш-функций в криптографии – обеспечение целостности данных. Они позволяют получить уникальный «цифровой отпечаток» (дайджест) информации. Любое, даже самое незначительное изменение исходных данных приведет к совершенно другому хэшу с высочайшей степенью вероятности. Это делает хэш-функции незаменимыми в следующих приложениях:

•Проверка целостности файлов и программного обеспечения.

•Электронная подпись (подписывается обычно не само сообщение, а его хэш).

•Аутентификация с помощью паролей (в базах хранятся только хэши паролей).

•Построение структур данных (например, хэш-таблицы, Merkle trees в блокчейне).

•Генерация псевдослучайных чисел и ключей.

1. Введение. Определение и назначение хэш-функций

В основе защиты данных с помощью хэш-функций лежит достаточно простая идея:

для любого массива данных мы можем вычислить некий эталон (хэш-код), и в дальнейшем в любой момент времени для данных может быть вычислен эталон повторно.

Если повторно вычисленный хэш-код, данных совпадает с эталонным – можно с высокой вероятностью утверждать что данные не были изменены.

Если повторно вычисленный хэш-код не совпал с эталонным – можно однозначно утверждать, что данные были изменены.

1. Введение. Определение и назначение хэш-функций

Математическая основа вычисления хэш-функций представлена в ГОСТ Р 34.11–2012.

Хэш-функция – функция, отображающая строки бит в строки бит фиксированной длины и удовлетворяющая следующим свойствам:

– по данному значению функции сложно вычислить исходные данные, отображаемые в это значение;

– для заданных исходных данных сложно вычислить другие исходные данные, отображаемые в то же значение функции;

– сложно вычислить какую-либо пару исходных данных, отображаемых в одно и то же значение.

1. Введение. Определение и назначение хэш-функций

Криптографическая хэш-функция – это математический алгоритм, который преобразует произвольный массив данных (сообщение) произвольной длины в битовую строку (хэш-код, дайджест) фиксированной длины.

Формально это можно записать как: **H(M) = h**, где:

- M – исходное сообщение произвольной длины.
- H – хэш-функция.
- h – хэш-код фиксированной длины (например, 256 бит для SHA-256).

1. Введение. Определение и назначение хэш-функций

В общем случае в информационных системах могут применяться различные алгоритмы получения хэш-функций для контроля целостности данных, такие как **MD5, SHA-1, SHA-2**, но в средствах криптографической защиты информации разрабатываемых и распространяемых на территории Российской Федерации хэш-функции рассчитываются по государственным стандартам, действующим на момент их разработки.

Лекция 2
Хэш-функции. Назначение и свойства хэш-функций (целостность, устойчивость к коллизиям).
Алгоритмы: SHA-256, SHA-3, ГОСТ Р 34.11-2012 ("Стрибог").

2. Криптографические свойства хэш-функций

2. Криптографические свойства хэш-функций

Рассмотрим требования, которым должна соответствовать хэш-функция для того, чтобы она могла использоваться в качестве **аутентификатора** сообщения.

Хэш-функция H, которая используется для аутентификации сообщений, должна обладать следующими свойствами:

1) Хэш-функция H должна применяться к блоку данных любой длины.
2) Хэш-функция H создает выход фиксированной длины.
3) H(M) относительно легко (за полиномиальное время) вычисляется для любого значения M.

Первые **три свойства** требуют, чтобы хэш-функция создавала хэш-код для любого сообщения.

2. Криптографические свойства хэш-функций

Свойство 4: Устойчивость к прообразу (Свойство односторонности)

Формулировка: Для заданного значения хэша h должно быть вычислительно невозможно найти **любое** исходное сообщение M', такое что H(M') = h.

Проще говоря: По отпечатку пальца невозможно восстановить человека. По хэшу невозможно восстановить исходные данные.

Значение: Это свойство обеспечивает защиту паролей. Если злоумышленник получит базу хэшей, он не сможет легко узнать сами пароли.

2. Криптографические свойства хэш-функций

Свойство 5: Устойчивость к второму прообразу (Слабая устойчивость к коллизиям)

Формулировка: Для заданного сообщения M1 должно быть вычислительно невозможно найти **другое** сообщение M2 (где $M1 \neq M2$), такое что $H(M1) = H(M2)$.

Проще говоря: Если у вас есть документ и его хэш, злоумышленник не должен суметь создать другой, поддельный документ с таким же хэшем.

Значение: Это свойство напрямую обеспечивает **целостность данных**. Гарантируется, что исходное сообщение не может быть незаметно подменено другим.

2. Криптографические свойства хэш-функций

Свойство 6: Устойчивость к коллизиям (Сильная устойчивость к коллизиям)

Формулировка: Должно быть вычислительно невозможно найти **любую пару** различных сообщений (M1, M2), таких что H(M1) = H(M2).

Важное замечание: Коллизии (когда два разных сообщения дают одинаковый хэш) **существуют** в силу того, что множество возможных сообщений бесконечно, а множество хэшей – конечно (из-за фиксированной длины). Задача криптографии – сделать поиск таких пар **практически невыполнимым** за разумное время.

Значение: Это свойство критически важно для безопасности алгоритмов электронной подписи. Если злоумышленник сможет найти две коллизирующие пары (например, безобидный договор и вредоносный), он может подписать безобидный вариант, а затем предъявить подпись для вредоносного.

2. Криптографические свойства хэш-функций

Хэш-функция, которая удовлетворяет первым пяти свойствам, называется **простой** или **слабой** хэш-функцией.

Если кроме того выполняется шестое свойство, то такая функция называется **сильной** хэш-функцией. Шестое свойство защищает против класса атак, известных как атака «день рождения».

Атака "день рождения" – это криптоаналитический метод, направленный на поиск коллизии для хэш-функции. Она использует тот же вероятностный принцип.

- Цель атаки: Найти два разных сообщения M1 и M2, таких что H(M1) = H(M2).
- Почему это возможно? Из-за ограниченности пространства хэшей. Хэш-функция с длиной выхода n бит может сгенерировать 2^n уникальных хэшей (например, для SHA-256 это 2^256 вариантов). Хотя это число огромно, оно конечно.

Лекция 2
Хэш-функции. Назначение и свойства хэш-функций (целостность, устойчивость к коллизиям).
Алгоритмы: SHA-256, SHA-3, ГОСТ Р 34.11-2012 ("Стрибог").

3. Обзор современных алгоритмов хэширования (алгоритмы: SHA-256, SHA-3, ГОСТ Р 34.11-2012 ("Стрибог")).

3. Хэш-функция MD4

Разработчик: Рональд Ривест (RSA Data Security, Inc.).

Архитектура: Использует конструкцию Меркла-Дамгора. Данные обрабатываются блоками по 512 бит, которые проходят три раунда (48 шагов) с различными нелинейными функциями.

Ключевые особенности:

- Вырабатывает хэш длиной **128 бит** (16 байт).
- Была разработана как быстрая и простая реализация хэш-функции (*одна из первых широко используемых хэш-функций).*
- Является прямым предшественником MD5 и более поздних хэш-функций (SHA-1, RIPEMD).
- **Безопасность:** **Взломан и абсолютно небезопасен.** Полностью уязвим к коллизиям (нахождению двух разных сообщений с одинаковым хэшем). В 2004 году была продемонстрирована практическая атака, позволяющая найти коллизию за долю секунды. **Использование MD4 запрещено в любых приложениях, требующих безопасности.**

Хронология появления MD4 или MD5

- **MD4** был разработан Рональдом Ривестом в **1990 году**.
- **MD5** был разработан Рональдом Ривестом в **1991 году** как прямая замена и усиленная версия MD4.
- Причиной для создания MD5 стало то, что в первом алгоритме очень быстро были обнаружены слабости. Ривест усложнил конструкцию MD4 (добавил четвертый раунд, дополнительные константы и т.д.), чтобы получить более стойкий алгоритм, который и был назван MD5.
- Таким образом, **MD5 является преемником и прямой эволюцией MD4**.

3. Хэш-функция MD5

Разработчик: Рональд Ривест (RSA Data Security, Inc.).
Архитектура: Использует конструкцию Меркла-Дамгора. Усложнена по сравнению с MD4: данные обрабатываются блоками по 512 бит, которые проходят четыре раунда (64 шага). Каждый раунд использует свою нелинейную функцию и добавляется уникальная аддитивная константа на каждом шаге.
Ключевые особенности:

- Вырабатывает хэш длиной **128 бит** (16 байт).
- Широко использовался для:
  - Проверки целостности данных (контрольные суммы файлов).
  - Хэширования паролей в базах данных (с использованием "соли").
  - Цифровых подписей и сертификатов.

3. Безопасность хэш-функция MD5

Безопасность: Взломан и не считается криптостойким!

Уязвим к коллизиям. В 2004 году была продемонстрирована практическая атака.

Существуют атаки, позволяющие создавать поддельные SSL-сертификаты и другие цифровые документы с валидной подписью.

Хотя поиск коллизий – быстрая задача, поиск прообраза (восстановление исходных данных по хэшу) все еще сложен, но не считается надежным.

Рекомендация: **Запрещен к использованию в любых новых криптографических системах!**

Существующие системы должны быть переведены на более безопасные алгоритмы, такие как SHA-256 или SHA-3.

3.Алгоритм SHA-256

Разработчик: NSA (Агентство национальной безопасности США), стандартизирован NIST.

Архитектура: использует **конструкцию Меркла-Дамгора**. Данные разбиваются на блоки, которые обрабатываются последовательно.

Ключевые особенности:

- Вырабатывает хэш длиной **256 бит** (32 байта).
- Широко распространен по всему миру. Является основой для:
  - Протоколов **SSL/TLS**.
  - Криптовалюты **Bitcoin**.
  - Многих систем контроля целостности ПО.

Безопасность: на текущий момент считается криптостойким и практичен для использования. Теоретически уязвим к атакам "длинного сообщения", но они не являются практичными для взлома.

3.Алгоритм SHA-3

Принят NIST в 2015 году как дополнение к SHA-2 (не замена).
Основан на алгоритме Keccak (победитель конкурса NIST 2007–2012).
Использует **«губчатую конструкцию» (sponge construction)** вместо классической схемы Меркла–Дэмгарда.
Поддерживает разные длины хэша: **224, 256, 384, 512 бит**.

Алгоритм можно использовать как хэш-функцию, потоковый шифр, для генерации случайных чисел.

Обеспечивает **устойчивость к коллизиям и предобразам**, в том числе против атак, эффективных на SHA-1/SHA-2.
Безопасность: считается перспективным для долгосрочной криптографической защиты.

3.Алгоритм ГОСТ Р 34.11-2012 ("Стрибог")

Разработчик: Российские криптографы (ВНИИСТ, ФСБ России). Принят как национальный стандарт РФ.

Архитектура: в основе лежит **отечественный блочный шифр "Кузнечик" (ГОСТ Р 34.12-2015)**. Алгоритм использует сеть Фейстеля.

Ключевые особенности:

- **Две модификации:** "Стрибог-256" (256 бит) и "Стрибог-512" (512 бит).
- **Нормативное требование:** является **обязательным** для использования в Российской Федерации в средствах криптографической защиты информации (СКЗИ) и для работы с квалифицированной электронной подписью (КЭП) согласно Федеральному закону № 63-ФЗ.

Назначение: защита информации в государственных информационных системах, обеспечение юридической значимости электронных документов в РФ.

3. Сравнение алгоритмов хэширования

Параметр	**SHA-256**	**SHA-3 (Keccak)**	**ГОСТ Р 34.11-2012 ("Стрибог")**
Страна/Стандарт	США (NIST)	США (NIST)	Россия
Длина выхода (бит)	256	224, 256, 384, 512	256, 512
Основа алгоритма	Конструкция Меркла-Дамгора (MDA)	Губчатая конструкция (sponge construction)	На основе блочного шифрования
Ключевые особенности	Часть семейства SHA-2, широкое распространение (SSL/TLS, Bitcoin)	Победитель конкурса NIST, принципиально новая конструкция для устойчивости к будущим атакам	Обязателен к использованию в РФ для защиты государственной информации и применения КЭП
Безопасность	Считается безопасным, но теоретически уязвим к атакам расширения длины	Устойчив к атакам, эффективным против SHA-2 (включая теоретические)	Соответствует российским стандартам безопасности, использует отечественные алгоритмы шифрования
Область применения	Международные системы, блокчейн, веб-безопасность	Перспективная замена SHA-2, где требуется устойчивость к новым угрозам	Государственные информационные системы, электронная подпись в РФ, СКЗИ

Выводы

Хэш-функции обеспечивают **целостность данных**, лежат в основе **электронной подписи** и многих механизмов аутентификации.

Выбор алгоритма зависит от нормативных требований и задач.

- **MD4 (MD5)** – исторические алгоритмы, использование которых запрещено. Могут применяться для задач на проверку целостности в невраждебных средах.
- **SHA-256** – мировой стандарт для большинства коммерческих приложений.
- **SHA-3** – перспективный и надежный алгоритм "на вырост".
- **ГОСТ Р 34.11-2012** – обязательный стандарт для использования в Российской Федерации в рамках выполнения требований ФЗ-63 "Об ЭП" и других нормативных актов.