Реферат на тему ‘Хеш-функция’

Основы информационной безопасности

Дворкина Ева Владимировна

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Изучить основные понятия, связанные с хеш-функциями, познакомиться с построением функции хеширования и алгоритмами хеширования данных.

2 Введение

В современном мире цифровой информации криптографические методы играют важнейшую роль в обеспечении безопасности данных. Среди этих методов хеш-функции занимают особое место. В этом реферате мы изучим понятие хеш-функции, основные свойства хеш-функций и различные алгоритмы, используемые для их реализации. Обсудим их роль в криптографии, применение в реальных системах.

3 Основные понятия

Хеширование (иногда хеширование, англ. hashing) – преобразование исходных данных (массив байтов) произвольной длины в выходные данные фиксированной длины.

Функции, выполняющие такие преобразования, называют **хеш-функциями** или функциями свёртки, входные данные – прообразом или **сообщением**, а результаты преобразования – **хешем, хеш-кодом, хеш-образом, хеш-суммой, контрольной суммой, цифровым отпечатком** или **дайджестом сообщения** (англ. message digest). [1]

3.1 Идеальная хеш-функция

Для идеальной хеш-функции выполняются следующие условия:

- **Детерминированность.** Хеш-функция является детерминированной, то есть одно и то же сообщение приводит к одному и тому же хеш-значению
- **Простота.** Значение хеш-функции быстро вычисляется для любого сообщения.
- **Необратимость.** невозможно найти сообщение, которое дает заданное хеш-значения. Для данного значения хеш-образа $h(m)$ сложно найти значение прообраза m. Это свойство также называют сопротивлением прообразу. Хеш-функция считается защищенной от нахождения прообраза, если существует очень низкая вероятность того, что злоумышленник найдет сообщение, которое сгенерировало заданный хеш.

- **Стойкость к коллизиям первого рода:** для заданного прообраза m_i сложно вычислить другой прообраз m_j, для которого значение хеш-образа совпадает $h(m_i) = h(m_j)$;
- **Стойкость к коллизиям второго рода:** сложно вычислить какую-либо пару прообразов m_i и m_j, для которых значения хеш-образов совпадают $h(m_i) = h(m_j)$ при $m_i \neq m_j$ Это свойство называют сопротивлением второму прообразу. Атака по нахождению второго прообраза происходит, когда злоумышленник находит определенный вход, который генерирует тот же хеш, что и другой вход, который ему уже известен. Другими словами, злоумышленник, зная, что $hash(m_1) = h$, пытается найти m_2 такое, что $hash(m_2) = h$
- **Отсутствие зависимости от отдельных элементов сообщения.** Небольшое изменение в сообщении изменяет хеш настолько сильно, что новое и старое значения кажутся некоррелирующими [4].

3.2 Коллизии

Коллизией хеш-функции $hash(m)$ называются два параметра a и b, при $hash(a) = hash(b)$. На практике это означает, что двум различным сообщениям соответствует одно и то же значение хеш-функции.

Коллизии существуют для большинства хеш-функций, но для «хороших» хеш-функций частота их возникновения близка к минимуму. Для хеш-функций, принимающих на вход сообщения переменной длины и возвращающих хеш постоянной длины, коллизии обязаны существовать, поскольку хотя бы для одного значения хеш-функции соответствующее ему множество входных данных будет бесконечно — и любые два набора данных из этого множества образуют коллизию.

Рассмотрим опасность возникновения коллизий на примере. В базах данных не рекомендуется хранить пароли пользователей в открытом виде. Вместо этого

обычно хранятся значения хеш-функций, полученные из этих паролей. Данный подход повышает безопасность, поскольку даже если злоумышленник получит доступ к хешу, он не сможет извлечь из него исходный пароль.

Однако, если злоумышленник способен найти сообщение, которое при применении известного алгоритма хеширования дает тот же хеш, что и исходный пароль, такое сообщение можно использовать для несанкционированного доступа к пользовательской учетной записи. Злоумышленник находит коллизии и может создавать поддельные сообщения, которые проходят проверку подлинности по хешу. Это может иметь серьезные последствия для безопасности системы, включая кражу учетных записей, подделку данных и другие виды мошенничества [3].

4 Семейства хеш-функций.

Правила идеальной хеш-функции особенно актуальны для криптографических хеш-функций, однако криптография - не единственная область применения хеш-функций. Рассмотрим различные семейства хеш-функций и их приминения.

4.1 CRC, HMAC и проверка контрольной суммы

Контрольная сумма представляет собой метод проверки целостности данных, используемый при передаче информации. При передаче данных по сети, например по протоколу TCP/IP, данные могут быть искажены из-за различных факторов, таких как помехи в канале связи или ошибки передачи.

Для предотвращения этой проблемы перед отправкой данных с помощью хеш-функции генерируется контрольная сумма на основе исходных данных. Контрольная сумма передается вместе с данными. При получении данных получатель рассчитывает контрольную сумму полученных данных с использованием того же алгоритма и сравнивает ее с переданной контрольной суммой.

Если контрольные суммы не совпадают, то данные были повреждены во время передачи. Получатель может запросить повторную отправку данных.

Семейство хеш-функций CRC (Cyclic Redundancy Check) предназначено для защиты данных от случайных искажений во время передачи данных. Эти функции не обеспечивают защиты от преднамеренных попыток взлома.

В отличие от CRC, хеш-функции HMAC (Hash-based Message Authentication Code) гарантируют, что данные не были изменены преднамеренно. HMAC использует криптографическую хеш-функцию (такую как SHA-256 или MD5) вместе с секретным ключом для создания аутентифицированной контрольной суммы [5].

4.2 Криптографические хеш-функции

4.2.1 Основной принцип работы

Хеш-функция принимает на вход сообщение M произвольной длины, которое представляется как массив байтов. Для удобства обработки сообщение дополняется нулевыми байтами до размера, кратного 64 битам. Это позволяет разделить сообщение на блоки по 64 бита. Каждый 64-битный блок делится на 16-битные подблоки. В зависимости от длины значения хеш-функции объявляется n переменных. В зависимости от конкретного алгоритма хеширования будут использоваться различные логические операции для изменения этих переменных, изначально содержащих константные значения. Эти операции включают различные логические операции между блоками сообщения и константами. В результате хеш-функция возвращает строку фиксированной длины, состоящую из суммы ранее объявленных переменных, обычно в шестнадцатеричном формате. Осуществляется отображение в меньшее пространство: Функция отображает эти входные данные в выходное значение гораздо меньшего фиксированного размера. Этот размер обычно составляет 128, 256 или 512 бит.

4.2.2 Семейства MD и SHA

Серия алгоритмов хеширования MD (MD1-MD6) была разработана Роном Райвестом, соавтором алгоритма RSA. MD2 стал первым широко используемым алгоритмом хеширования в криптографии. MD4 был популярен на протяжении

длительного времени, впоследствии его заменил MD5, который пользовался значительной популярностью. Однако в конце 1990-х годов были обнаружены предпосылки для взлома MD5, и в настоящее время он признан небезопасным. . Алгоритм MD6 участвовал в конкурсе SHA-3, но не получил широкого распространения. Алгоритмы линейки SHA (SHA-1, SHA-2, SHA-3) являются широко используемыми в настоящее время алгоритмами хеширования. Они имеют большую длину хеш-функции и, следовательно, работают медленнее, чем алгоритмы MD. Наблюдается переход от SHA-1 к стандартам SHA-2 и SHA-3. SHA-2 объединяет алгоритмы SHA224, SHA256, SHA384 и SHA512. Они используются в криптовалютах (например, Tor и Bitcoin). Стандарт SHA-2 был разработан в 2002 году, и до 2008 года в нем не обнаруживались коллизии. После их обнаружения был запущен конкурс на разработку стандарта SHA-3, в котором победил алгоритм Keccak. В настоящее время Keccak используется, например, в криптовалютах. Наиболее популярными криптографическими хеш-функциями на текущий момент являются SHA-2 и SHA-3.

4.2.3 ГОСТ Р 34.11-2012 для цифровых подписей

ГОСТ Р 34.11-2012 (неофициальное название — Стрибог) — текущий российский криптографический (стойкий к взлому) алгоритм введенный в работу в 2013 году (ранее использовался ГОСТ Р 34.11-94). Длина выходного хеша может быть 256 или 512 бит. Обладает высокой криптостойкостью и довольно хорошей скоростью работы. Используется для электронных цифровых подписей преимущественно в системе государственного документооборота [6].

5 Применение хеш-функций

Рассмотрим несколько простых примеров применения хеш-функций:

- Проверка целостности сообщений и файлов. Сравнивая хеш-значения сообщений, вычисленные до и после передачи, можно определить, были ли внесены какие-либо изменения в сообщение или файл.
- Верификация пароля. Проверка пароля использует криптографические хеши. Хранение всех паролей пользователей в виде открытого текста может быть небезопасно, если файл паролей будет украден. Одним из способов уменьшения этой опасности является хранение в базе данных не самих паролей, а их хешей. При выполнении хеширования исходные пароли не могут быть восстановлены из сохраненных хеш-значений.
- Цифровая подпись. Подписываемые документы имеют различный объем, поэтому в схемах электронных подписей подпись ставится не на сам документ, а на его хеш. Вычисление хеша позволяет выявить малейшие изменения в документе при проверке подписи.

Другие области применения хеш-функций:

- ускорение поиска данных с использование хеш-таблиц и хеш-карт;
- поиск дубликатов или аналогов исходных данных при условии, что коллизии минимальны;
- защита данных от изменения (блокчейн);
- доказательство работы – затрат вычислительных ресурсов (биткоин, другие криптовалюты) [4].

6 Заключение

Хеш-функции являются важными инструментами в криптографии и информационной безопасности. Они используются для обеспечения целостности и подлинности данных, создания цифровых подписей и других задач.

7 Выводы

Познакомились с понятием хеш-функции и рассмотрели основные принципы ее работы и построения, а также с некоторыми алгоритмами хеширования.

Список литературы

1. Охотин А. Хэш-функции. Хэш-таблицы [Электронный ресурс]. 2022. URL: https://users.math-cs.spbu.ru/~okhotin/teaching/algorithms3_2022/okhotin_algorithms3_2022_l4_5.pdf.
2. lec10 (Хеш-функции) [Электронный ресурс]. URL: https://esystem.rudn.ru/mod/folder/view.php?id=1102741.
3. Хеш-функция, что это такое? [Электронный ресурс]. 2020. URL: https://habr.com/ru/articles/534596/.
4. Викторович А.В. Хеш-функции. [Электронный ресурс]. URL: https://sites.google.com/site/anisimovkhv/learning/kripto/lecture/tema9.
5. Хэш-алгоритмы [Электронный ресурс]. 2010. URL: https://habr.com/ru/articles/93226/.
6. КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ. Функция хеширования [Электронный ресурс]. 2012. URL: https://yztm.ru/wp-content/uploads/2018/03/gost-34.11-2012.pdf?ysclid=lvpzek763j909219784.