Хеш функції.

Ігнатенко О.І.

Подходы к контролю целостности данных

- Контрольная сумма
- Выработка **MDC** – ***Manipulation Detection Code*** – *кода обнаружения манипуляций* (с данными) - Хеш-функция (message digest)
- Выработка **MAC** – ***Message Authentification Code*** – *кода аутентификации сообщений* – электронно-цифровая подпись

Расшифровка номера карты

Первая цифра — это основной идентификатор индустрии.

- 1, 2 — авиакомпании;
- 3 — туризм, развлечения;
- 4, 5 — финансовые организации, банки;
- 6 — торговля, банковская сфера;
- 7 — нефтяные компании;
- 8 — телекоммуникации;
- 9 — государственные предприятия.

Первая цифра — указывает на платежную систему.

- American Express (34xxxx, 37xxxx)
- VISA (4xxxxx)
- MasterCard (51xxxx - 55xxxx)
- Discover (6011xx, 644xxx, 65xxxx)
- Maestro (3-, 5-, 6-)

Расшифровка номера карты

Первые шесть цифр представляют собой идентификационный номер эмитента. Другими словами, данный номер обозначает компанию, выдавшую карточку и тип карты.

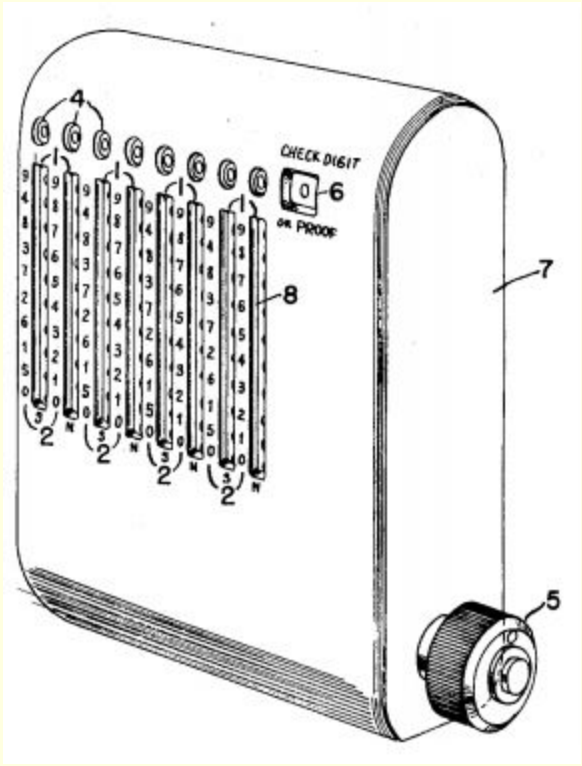
Девять цифр, начиная с седьмой, представляют собой идентификационный код держателя данной пластиковой карты.

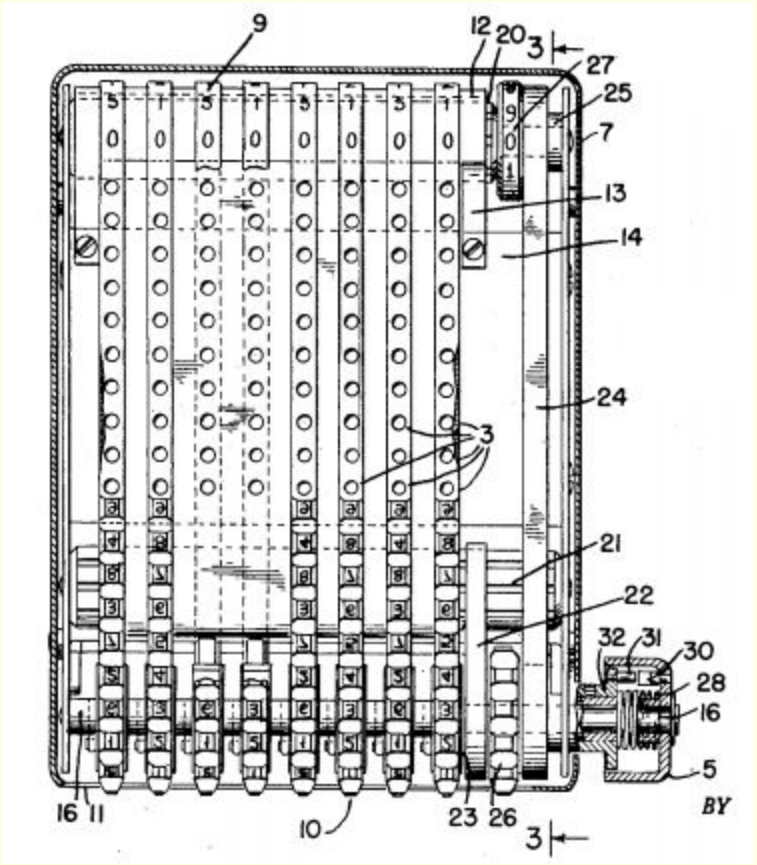
А вот самая последняя цифра предназначена для контроля (алгоритм Луна)

Алгоритм Луна

- Алгоритм вычисление контрольной суммы
- Стандарт ISO/IEC 7812
- Описан в 1954, патент 1960
- Сотрудник IBM Ганс Питер Лун
- Алгоритм не криптографический, реализует проверку целостности
- Предназначен для выявления ошибок либо искажений при вводе номеров:
  - кредитных карт
  - номеров дисконтных карт
  - кодов соц. страхований
  - IMEI - кодов

Запатентованная реализация (1960)





Суть алгоритма Луна

1. Цифры проверяемой последовательности нумеруются справа налево.
2. Цифры, оказавшиеся на нечётных местах, остаются без изменений.
3. Цифры, стоящие на чётных местах, умножаются на 2.
4. Если в результате такого умножения возникает число больше 9, оно заменяется суммой цифр получившегося произведения — однозначным числом, т.е. цифрой.
5. Все полученные в результате преобразования цифры складываются. Если сумма кратна 10, то исходные данные верны.

Односторонняя хэш-функция:

Message digest cryptographic checksum

Исходные данные: сообщение M любой длины; h - последовательность фиксированной длины.

1. по M легко вычисляется h(M)
2. по h трудно найти M
3. по M трудно найти M′ : h(M) = h(M′).
4. Желательно чувствительность к не значительным изменениям

Три способа разработки:

- на основе труднорешаемой мат. задачи;
- на основе алгоритмов блочного шифрования (DES);
- самостоятельная разработка.

Практическая реализация

- MD5
- SHA-1
- SHA-3 (Keccak)
- 128-160-224/256/384/512 bit результат
- Используют, как правило, два входа: исходное сообщение и предыдущий хеш

Secure Hash Algorithm (SHA)

- Rfc 3174
- 160-bit результат
- Если исходный текст <2^64, то формирует 160-bit хеш

Описание алгоритма SHA

1. Дополнение блоков (padding)
2. Инициализация 5-ти переменных (MD5-4)
3. Основной итерационный блок (512 бит)
    1. Подстановка переменных
    2. 4 раунда по 20 операций (MD5-16). Нелинейные операции, сдвиги и сложения
    3. Расширение сообщения.
    4. Сложение результата с промежуточным результатом
4. Переход к следующему блоку

Этапы алгоритма SHA

1. Дополнение блоков (padding)

В начале, сообщение достраивается до длины, кратной 512 (padding): к сообщению добавляется «1», затем столько нулей, сколько надо до длины, кратной 512 минус 64 бита, затем 64-битное представление длины сообщения до padding'а (алгоритм padding'а такой же, как в MD5)

2. Инициализация 5-ти переменных (MD5-4)

A=67 45 23 01

B=EF CD AB 89

C=98 BA DC FE

D=10 32 54 76

E=C3 D2 E1 F0.

Основной итерационный блок алгоритма SHA

1. Подстановка переменных
   - AA=A; BB=B; CC=C; DD=D; EE=E;
2. 4 раунда по 20 операций (MD5-16). Нелинейные операции, сдвиги и сложения

В алгоритме также используется 4 константы:

$K_t = 5A827999$	первые 20	$f_1(x,y,z) = ((x \,\&\, y) \mid (\sim x \,\&\, z))$
$K_t = 6ED9EBA1$	вторые 20	$f_2(x,y,z) = (x \hat{} y \hat{} z)$
$K_t = 8F1BBCDC$	третьи 20	$f_3(x,y,z) = ((x \,\&\, y) \mid (x \,\&\, z) \mid (y \,\&\, z))$
$K_t = CA62C1D1$	четвертые 20	$f_4(x,y,z) = (x \hat{} y \hat{} z)$

Основной итерационный блок алгоритма SHA

3. Расширение сообщения.

Блок сообщения преобраз. из 16 32-битн. слов ($M_0,..,M_{15}$) в 80 32-битн. слов ($W_0,...,W_{79}$) по следующему алгоритму:

$$W_t = M_t, t = 0..15$$

$$W_t = W_{t-3}\ XOR\ W_{t-8}\ XOR\ W_{t-14}\ XOR\ W_{t-16}, t = 16..79$$

Если t – номер операции (1..80) M_j -- j-ый подблок сообщения (0..15), <<<S – циклический сдвиг влево на S бит, то 80 операций выглядят как:

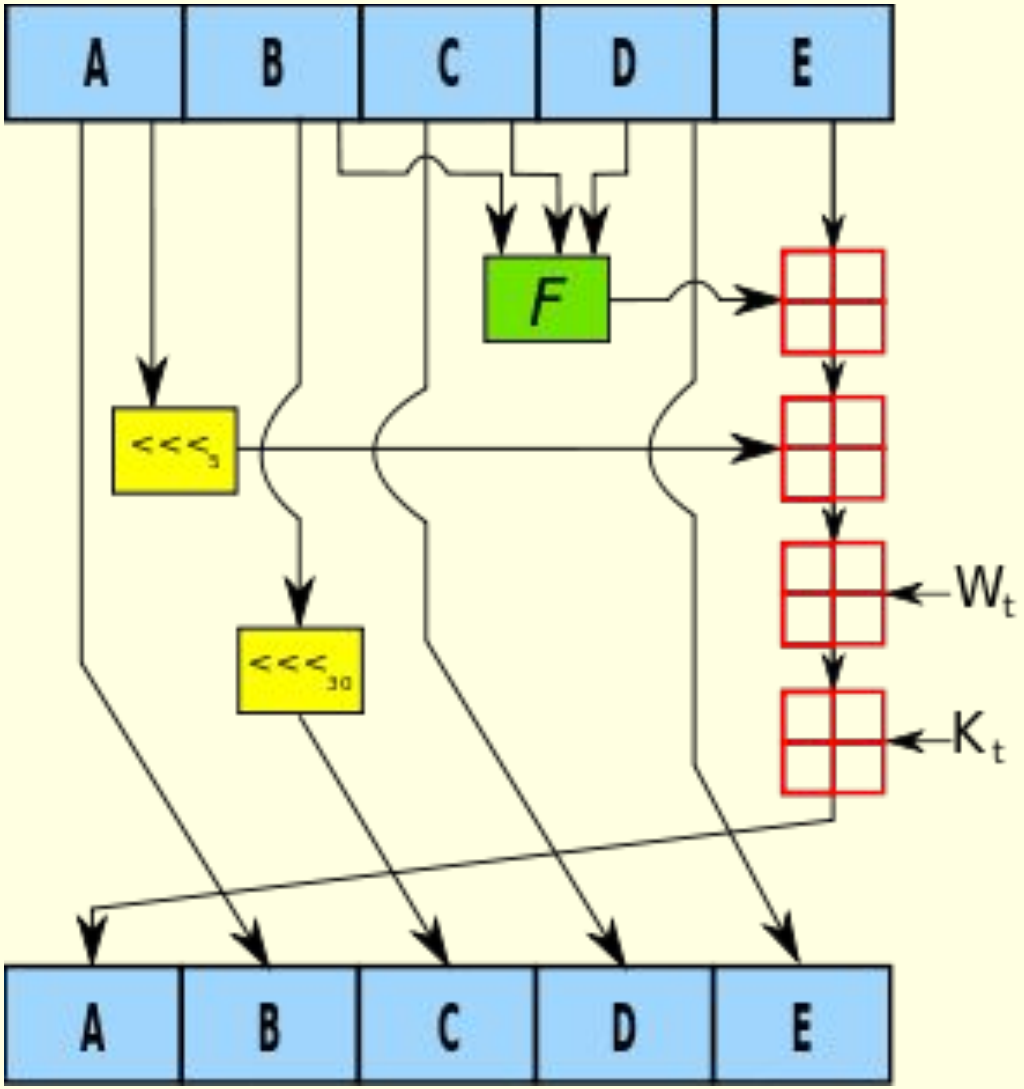
TEMP=(A<<<5) + $f_t(B,C,D) + E + W_t + K_t$

E = D
D = C
C = B<<<30
B = A
A = TEMP

4. Сложение результата с промежуточным результатом

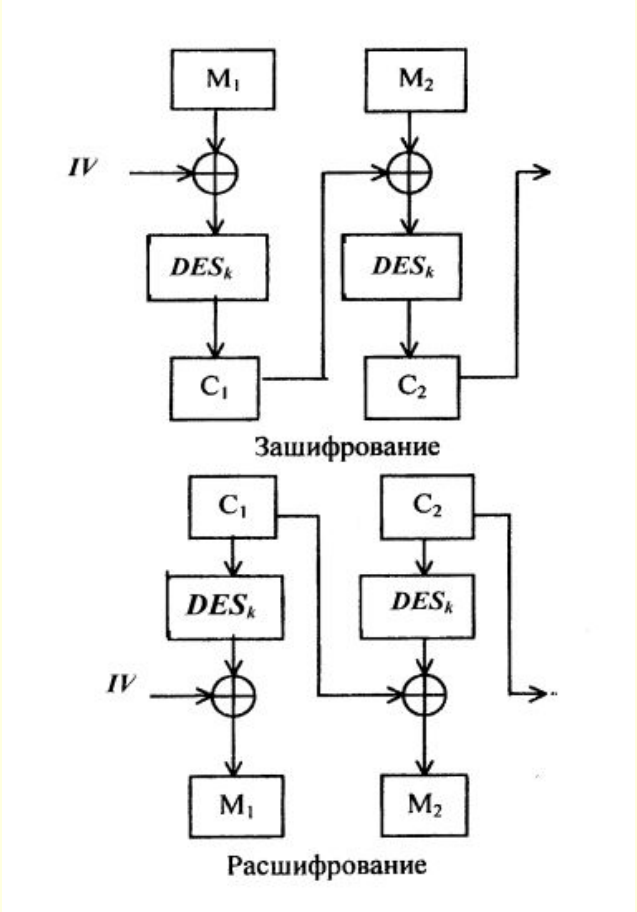
Далее A, B, C, D, E прибавляют к AA, BB, CC, DD, EE и берется следующий блок. Результат– конкатенация AA, BB, CC, DD, EE.

Схема выполнения одной операции SHA



Использование блочных алгоритмов

- Использование DES в режиме CBC (Режим сцепления блоков), CFB (Режим обратной связи по шифрограмме) с фиксированным ключом.
- Использование одного ключа



- Xor криптограммы предыдущего и текста текущего

Метод увеличения хэша

Для получения хэш-значения с большей длиной, чем это позволяет выбранная хэш-функция, был предложен следующий метод:

1) Сгенерировать хэш от сообщения
2) Добавить хэш в конец сообщения (append)
3) Сгенерировать хэш от конкатенации сообщения и хэша
4) Получить длинное хэщ-значение путем конкатенации хэш-значения из пункта 1) и хэш-значения из пункта 3)
5) Повторять шаги (1-3) до получения необходимой длины.

Надежность или ненадежность данного метода не доказаны.

Криптоанализ

1. «Лобовая атака»

Имея h(M), найти M1, такую что h(M1). – 2^160

2. «Парадокс дней рождения» - 2^80

(1) Аня готовит 2 версии контракта – один выгодный для Вани, а другой разоряющий его;

(2) Аня делает несколько малозначительных изменений к каждому документу и считает хэш-значения каждый раз (напр., заменить SPACE на SPACE-BACKSPACE-SPACE, добавить один или два пробела перед переводом строки и т.д. Только путем одного «изменения / оставления как было» на каждой строке Аня может сгенерировать 32^2 документов).

(3) Аня сравнивает набор хэшей для обоих документов, подыскивая одинаковые пары (если х.-ф. = 64 бит, то обычно хватит 32^2 пар). Выбирается та пара, что дает одинаковые хэши.

(4) Аня подсовывает Ване выгодный контракт; Ваня подписывает его хэш.

(5) Аня может теперь доказать, что Ваня подписал невыгодный ему контракт.

Применение

- Контрольная сумма

Проверка целостности сообщения (файла)

- Элемент цифровой подписи

(Аутентификация источника данных)

- Элемент аутентификации

Задачи

1. Реализовать свою хеш-функцию с результатом в 4 бита
2. Оценить сложность подделки документа word для получения такого дайжест сообщения

Источники

- *Романец Ю.В., Тимофеев П.А. Шаньгин В.Ф. Защита информации в компьютерных системах и сетях. – 2-е изд. перераб. и доп. – М.: 2001. – 276с.*
- *Венбо Мао. Современная криптография. Теория и практика = Modern Cryptography: Theory and Practice. — М.: Вильямс, 2005. — 768 с.*
- *Шнайер Б.* Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си = Applied Cryptography. Protocols, Algorithms and Source Code in C. — М.: Триумф, 2002. — 816 с.
- Ян С. Й. Криптоанализ RSA. — М.—Ижевск: НИЦ «Регулярная и хаотическая динамика», Ижевский институт компьютерных исследований, 2011. — 312 с.
- http://protect.htmlweb.ru/ecp.htm