Однонаправленные хэш-функции

1. Цель работы

Изучить и реализовать различные алгоритмы однонаправленного хэширования данных.

2. Указания к работе

Однонаправленная функция $H(M)$ применяется к сообщению произвольной длины M и возвращает значение фиксированной длины h.

$h = H(M)$, где h имеет длину m

Многие функции позволяют вычислять значение фиксированной длины по входным данным произвольной длины, но у однонаправленных хэш-функций есть дополнительные свойства, делающие их однонаправленными

Зная M, легко вычислить h.

Зная H, трудно определить M, для которого $H(M)=h$.

Зная M, трудно определить другое сообщение, M', для которого $H(M)= H(M')$.

В некоторых приложениях однонаправленности недостаточно, необходимо выполнение другого требования, называемого **устойчивостью к столкновениям**.

Должно быть трудно найти два случайных сообщения, M и M', для которых $H(M)= H(M')$.

Пример использования неустойчивости к столкновениям:

(1) Алиса готовит две версии контракта: одну, выгодную для Боба, и другую, приводящую его к банкротству

(2) Алиса вносит несколько незначительных изменений в каждый документ и вычисляет хэш-функции . (Этими изменениями могут быть действия, подобные следующим : замена ПРОБЕЛА комбинацией ПРОБЕЛ-ЗАБОЙ-ПРОБЕЛ, вставка одного-двух пробелов перед возвратом каретки, и т.д. Делая или не делая по одному изменению в каждой из 32 строк, Алиса может легко получить 2^{32} различных документов.)

(3) Алиса сравнивает хэш-значения для каждого изменения в каждом из двух документов , разыскивая пару, для которой эти значения совпадают. (Если выходом хэш-функции является всего лишь 64-разрядное значение, Алиса, как правило, сможет найти совпадающую пару сравнив 2^{32} версий каждого документа.) Она восстанавливает два документа, дающих одинаковое хэш-значение .

(4) Алиса получает подписанную Бобом выгодную для него версию контракта, используя протокол, в котором он подписывает только хэш-значение .

(5) Спустя некоторое время Алиса подменяет контракт, подписанный Бобом, другим, который он не подписывал. Теперь она может убедить арбитра в том, что Боб подписал другой контракт .

Как видно из примера, для нахождения двух документов с равными хэш-значениями, длинною 64 бита надо перебрать 2^{32} документов.

Для удлинения хэн-значений, выдаваемых конкретной хэш-функцией, был предложен следующий метод .

(1) Для сообщения с помощью одной из упомянутых в этой книге однонаправленных хэш-функций генерируется хэш-значение.

(2) Хэш значение добавляется к сообщению .

(3) Генерируется хэш-значение объединения сообщения и хэш-значения этапа (1).

(4) Создается большее хэш-значение, состоящее из объединения хэш-значения этапа (1) и хэш-значения этапа (3).

(5) Этапы (1)-(4) повторяются нужное количество раз для обеспечения требуемой длины хэш-значения .

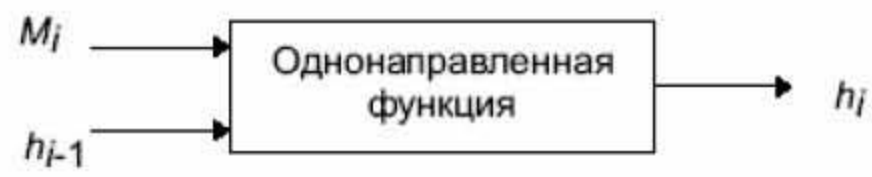
Обзор однонаправленных хэш-функций

На практике однонаправленная хэш-функция строится на идее **функции сжатия.** Такая однонаправленная функция выдаёт хэш-значение длинны n при заданных входных данных длинны m большей n. Входами функции сжатия являются блок сообщения и выход предыдущего блока текста.

То есть, хэш-значение блока M_i равно

$h_i = f(M_i, h_{i-1})$

Это хэш-значение вместе со следующим блоком сообщения становится следующим входом функции сжатия . Хэш-значением всего сообщения является хэш-значение последнего блока .



Хэшируемый вход должен каким-то способом содержать бинарное представление длины всего сообщения . Таким образом преодолевается потенциальная проблема, вызванная тем, что сообщения различной длины могут давать одно и то же хэш-значение

Полезной мерой для хэш-функций, основанных на блочных шифрах, является **скорость хэширования**, или количество *n*-битовых блоков сообщения (*n* - это размер блока алгоритма), обрабатываемых при шифровании .

Однонаправленные хэш-функции, использующие симметричные блочные алгоритмы

В качестве однонаправленных хэш-функций можно использовать симметричные блочные алгоритмы ши фрования. Идея в том, что если безопасен блочный алгоритм, то и однонаправленная хэш-функция будет без опасной.

Самым очевидным способом является шифрование сообщения в режиме CBC или CFB с помощью фиксированного ключа и IV, хэш-значением будет последний блок шифротекста . Эти методы описаны в различных стандартах, использующих DES:
1)CBC,
2)CFB
Их описание можно найти в л.р №5.

Способ поумнее использует в качестве ключа блок сообщения , предыдущее хэш-значение в качестве входа, а текущее хэш-значение служит выходом .

Действительные хэш-функции даже еще сложнее. Размер блока обычно совпадает с длиной ключа, и размером хэш-значения будет длина блока. Так как большинство блочных алгоритмов 64-битовые , спроектирован ряд схем, дающих хэш-значение в два раза большее длины блока .

При условии, что хэш-функция правильна , безопасность этой схемы основана на безопасности используемой блочной функции. Однако есть и исключения. Дифференциальный криптоанализ лучше работает против блочных функций в хэш-функциях, чем против блочных функций, используемых для шифрования : ключ известен, поэтому можно использовать различные приемы. Для успеха нужна только одна правильная пара, и можно генерировать столько выбранного открытого текста, сколько нужно .

Схемы, в которых длина хэш-значения равна длине блока

Вот общая схема (см. рис. 1):

$H_0 = I_H$, , где I_H - случайное начальное значение

$H_i = E_A(B) \oplus C$

где A, B и C могут быть либо M_i, H_{i-1}, $(M_i \oplus H_{i-1})$, либо константы (возможно равные 0). H_0 - это некоторое случайное начальное число I_H. Сообщение разбивается на части в соответствии с размером блока , M_i, обрабатываемые отдельно.

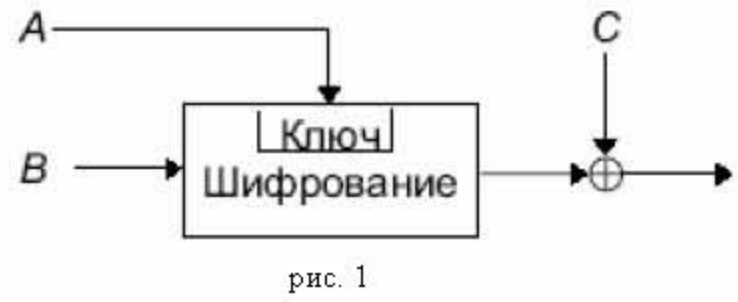


рис. 1

Таких комбинаций A, B и C может быть всего 64.

Ниже приведены четыре самых безопасных хэш-функции.

1)

3)

2)

4)

Схемы в которых длинна хэш-значения равна удвоенной длине блока

Preneel-Bosselaers-Govaerts-Vandewalle

При 64-битовом блочном алгоритме схема выдает два 64-битовых хэш-значения , G_i и H_i, объединение которых и дает 128-битовое хэш-значение. У большинства блочных алгоритмов длина блока равна 64 битам . Два соседних блока, L_i и R_i, размер каждого равен размеру блока, хэшируются вместе.

$G_0 = I_G$, где I_G - случайное начальное значение

$H_0 = I_H$, , где I_H - другое случайное начальное значение

$$G_i = E_{L_i \oplus H_{i-1}}(R_i \oplus G_{i-1}) \oplus R_i \oplus G_{i-1} \oplus H_{i-1}$$

$$H_i = E_{L_i \oplus R_i}(H_{i-1} \oplus G_{i-1}) \oplus L_i \oplus G_{i-1} \oplus H_{i-1}$$

Quisquater-Girault

Эта схема генерирует хэш-значение, в два раза большее длины блока. Ее скорость хэширования равна 1 . Она использует два хэш-значения, G_i и H_i, и хэширует вместе два блока, L_i и R_i.

$G_0 = I_G$, где I_G - случайное начальное значение

$H_0 = I_H$, , где I_H - другое случайное начальное значение

$$W_i = E_{L_i}(G_{i-1} \oplus R_i) \oplus R_i \oplus H_{i-1}$$

$$G_i = E_{R_i}(W_i \oplus L_i) \oplus G_{i-1} \oplus H_{i-1} \oplus L_i$$

$$H_i = W_i \oplus G_{i-1}$$

3. Варианты заданий

Номер варианта	Схема шифрования	Алгоритм шифрования
1	(*хэш-значение* =*длина блока*) ; **схема №1**	ГОСТ
2	(*хэш-значение* =*длина блока*) ; **схема №2**	DES
3	(*хэш-значение* =*длина блока*) ; **схема №3**	**гаммирование**
4	(*хэш-значение* =*длина блока*) ; **схема №4**	ГОСТ
5	***Preneel-Bosselaers-Govaerts-Vandewalle***	DES
6	***Quisquater-Girault***	**гаммирование**
7	***Quisquater-Girault***	ГОСТ
8	***Quisquater-Girault***	DES
9	(*хэш-значение* =*длина блока*) ; **схема №2**	**гаммирование**
10	(*хэш-значение* =*длина блока*) ; **схема №3**	ГОСТ
11	(*хэш-значение* =*длина блока*) ; **схема №4**	DES
12	(*хэш-значение* =*длина блока*) ; **схема №1**	**гаммирование**
13	***Preneel-Bosselaers-Govaerts-Vandewalle***	ГОСТ
14	***Quisquater-Girault***	DES
15	***Preneel-Bosselaers-Govaerts-Vandewalle***	**гаммирование**

4. Контрольные вопросы

1) Что такое хэш-функция, для чего их используют?
2) Свойство односторонности хэш-функции и его значение.
3) В чём заключается устойчивость к столкновениям, на что она влияет?
4) Что и сколько раз надо сделать чтобы обмануть подписчика?
5) Как увеличить длину хэш-значения?
6) Хэширование с помощью блочных алгоритмов.
7) Схема хэширования с длиной хэш-значения равной длине блока. Преимущества. Недостатки.
8) Схема хэширования с длиной хэш-значения равной длине удвоенной длине блока. Преимущества. Недостатки.