ХЕШ-ФУНКЦІЇ ТА ЇХ ВЛАСТИВОСТІ

Визначення. **Криптографічна хеш-функція** (*"hash" – мішанина*) **–** це функція (в загальному випадку – алгоритм), що перетворює вхідні дані **будь-якого розміру** в дані **фіксованого** розміру.

Найбільш поширені хеш-функції:

MD4, MD5

SHA1, SHA256, SHA512, SHA3-256, SHA3-512

RIPEMD - 160

Результат хешування вхідного потоку байт – послідовність визначеної довжини, яку називають **хеш-кодом**, **хеш-сумою**, **хешем** або **дайджестом повідомлення**.

№	Вхідні дані, *M*	*hash*	Результат, *h(M)*
1	*'Hello, world!'*	sha1	943a702d06f34599aee1f8da8ef9f7296031d699
2	*'12345'*	sha1	8cb2237d0679ca88db6464eac60da96345513964
3	*'abs'*	sha1	a9993e364706816aba3e25717850c26c9cd0d89d
4	''	sha1	da39a3ee5e6b4b0d3255bfef95601890afd80709
5	*'Marchenko Vladyslav'*	md5	20dc1d87a408f05cc319af33f7fe94b3
6	*'Marchenko Vladyslav'*	sha1	6d2d4a56524c214d6bbe712d09e253575c34c9d7
7	*'Marchenko Vladyslav'*	sha224	34b14b55461f8e4af92bd4bd5ed5e07244eb63a7dc 9e02b70bfcc90f
8	*'Marchenko Vladyslav'*	sha256	bbea36ac29a7ca982cb46b906f3e9efaf3fe077a6a7c abb2cc0a93698dd67e88
9	*'Marchenko Vladyslav'*	sha512	c65d30cdf0d0bed87fdfb8f61e65199a3dfdda0f279ddc cd00d3bf309a9a8123befde92a7d14d174470b9f0f473 e9e3ecc3f4ab69d682a7024cace4234c4e595
10	*zayava.docx*	sha1	389a533e3fe54ce16bfef594d4b96fcc892ba721

Вимоги до криптографічних хеш-функцій

Метою функції хешування є отримання «дактилоскопічної» характеристики файлу, повідомлення або взагалі будь-якого блоку даних. Щоб бути корисною для аутентифікації повідомлень, функція хешування $H(x)$ повинна мати такі властивості:

1. Бути застосовною до блоку даних будь-якої довжини.

2. Давати на виході значення фіксованої довжини.

3. Значення $H(x)$ має обчислюватися відносно легко для будь-якого заданого x, а алгоритм обчислення має бути практичним з погляду як апаратної, так і програмної реалізації.

4. **Односторонність (незворотність).** Для будь-якого даного коду h повинно бути практично неможливо вирахувати x, для якого $H(x) = h$.

5. **Стійкість до *колізій* першого роду.** Для будь-якого заданого блоку x має бути практично неможливо обчислити $y \neq x$, для якого $H(x) = H(y)$.

6. **Стійкість до *колізій* другого *роду*.** Має бути обчислювально неможливо підібрати пару повідомлень x , y, для яких $H(x) = H(y)$.

7. **Лавиновий ефект**.

Здатність функції хешування протистояти атакам з перебором варіантів залежить виключно від довжини хеш-коду, що породжується алгоритмом. Для хеш-коду довжини n порядок необхідних зусиль пропорційний наступним величинам:

Односторонність	2^n
Стійкість до *колізій* першого роду	2^n
Стійкість до *колізій* другого *роду*	$2^{n/2}$

Загальні принципи побудови хеш-функцій

1. Прості алгоритми шифрування без використання секретного ключа, що працюють в режимі CBC.

Початкове повідомлення M розбивається на блоки M_1, M_2, …, M_N. За допомогою будь-якого традиційного симетричного алгоритму, наприклад, за допомогою DES, для отримання хеш-коду G поступово обчислюються:

H_0 = *початкове значення*

$H_1 = E_{M1}(H_0)$

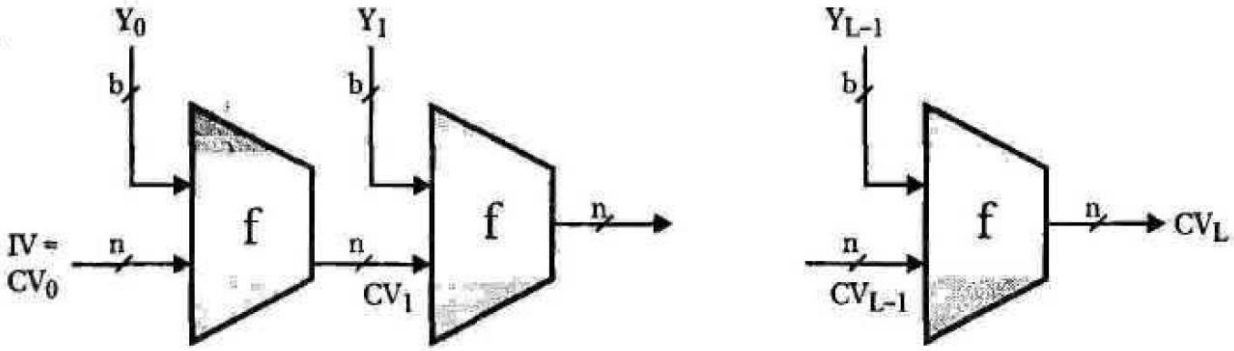
…

$Hi = E_{Mi}(H_{i-1})$

…

$G = H_N$

2. Застосування **функції стиску** *f*.



IV – початкове значення (Initial Vector)

CV_i *– змінна зчеплення*

Y_i *– i-ий блок вхідного повідомлення (i = 0 .. L-1)*

f – алгоритм стиску, який буде заснований L разів

L – кількість блоків, на яку ділиться вхідне повідомлення

n – довжина хеш-коду

b – довжина вхідного блоку Y_i (*як правило,* $b > n$).

Функція хешування отримує на вхід повідомлення M і ділить його на L блоків рівної фіксованої довжини по b бітів кожен. Якщо необхідно, останній блок доповнюється до b бітів. В останній блок також включається значення сумарної довжини вхідного повідомлення. Це робить завдання супротивника ще складнішим.

Алгоритм хешування передбачає багаторазове застосування **функції стиску** f , що отримує на вхід два значення: n-бітове значення, отримане на попередньому етапі – змінну зчеплення, і b-бітовий блок вхідного повідомлення) і породжує n-бітове результуюче значення. На початку хешування змінна зчеплення отримує початкове значення CV_0, що є частиною алгоритму. Кінцеве значення змінної зчеплення CV_L і буде значенням функції хешування. Зазвичай $b > n$, тому й говорять про алгоритм стиску.

Функція хешування може бути формально описана наступним чином:

$CV_0 = IV$ – початкове n-бітове значення

$CV_i = f(CV_{i-1}, Y_i), \quad 1 \le i \le L,$

$H(M) = CV_L$,

де $Y_0, Y_1, \ldots, Y_{L-1}$ – b-бітові блоки, на які поділяється вхідне повідомлення M.