Московский институт электроники и математики им. А.Н. Тихонова

Кафедра информационной безопасности киберфизических систем

Москва 2024

Криптографические методы защиты информации

ГОСТ Р 34.11-2012

Хэширование

Хэширование

- **Хэширование** — это преобразование входной битовой строки произвольной длины в выходную битовую строку фиксированной длины.
- Хэш-функция: сообщение → хэш-код.
- Хэширование предназначено для обеспечения контроля целостности информации.

Пример хэширования текста

Национальный исследовательский университет «Высшая школа экономики» — исследовательский университет, осуществляющий свою миссию через научно-образовательную, проектную, экспертно-аналитическую и социокультурную деятельности на основе международных научных и организационных стандартов.	Национальный исследовательский университет «Высшая Школа Экономики» — исследовательский университет, осуществляющий свою миссию через научно-образовательную, проектную, экспертно-аналитическую и социокультурную деятельности на основе международных научных и организационных стандартов.
6B4FD9F1D385D9FCCD7532ED1254B7B44EA1FBF76F8F9766DAE79F20C83AA619	F8A42D79D2AE48D6448CE589720A39E828277A76F35CD10B41F46DD75C6C20CB

Свойства криптографических хэш-функций

- Защищенность от восстановления прообразов.
- Защищенность от коллизий.
- Защищенность от вторых прообразов.
- Лавинный эффект.

- **Восстановление прообразов** – по известному хэш-коду h найти такое сообщение m, что $h(m) = h$.
- **Коллизия** – ситуация, когда для некоторых различных сообщений $m_1 \neq m_2$ хэш-коды одинаковы: $h(m_1) = h(m_2)$.
- **Второй прообраз** – для данной пары (сообщение m_1, хэш-код $h = h(m_1)$) такое сообщение m_2, что $m_1 \neq m_2$ и $h(m_1) = h(m_2)$.

Если $h(\acute{M}) = h(M)$, то $\acute{M} = M$

Если $h(\acute{M}) \neq h(M)$, то $\acute{M} \neq M$

Отечественные стандарты хэширования

- **ГОСТ Р 34.11–94**. Информационная технология. Криптографическая защита информации. Функция хэширования. (устаревший):
  - Длина хэш-значения: 256 бит.
  - Длина блока данных: 256 бит.
  - Использует симметричный шифр ГОСТ 28147–89.
- **ГОСТ Р 34.11–2012**. Информационная технология. Криптографическая защита информации. Функция хэширования.
  - Длина хэш-значения: 256 бит, 512 бит.
  - Длина блока данных: 512 бит.
  - Не использует других криптографических алгоритмов.

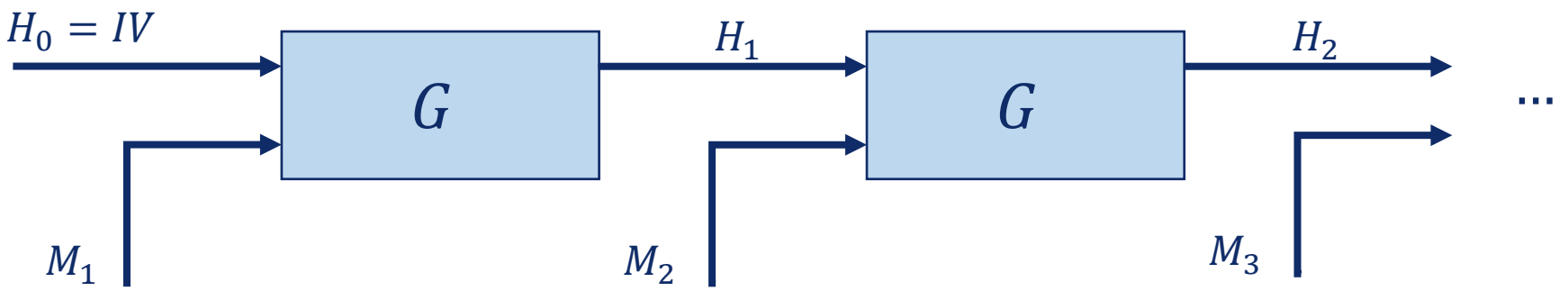
Зарубежные стандарты хэширования

Наименование	Год разработки	Длина хэш-кода	Стандарт
MD2 (The MD2 Message-Digest Algorithm)	1989	128	RFC 1319
MD4 (Message Digest 4)	1990	128	RFC 1186
MD5 (Message Digest 5)	1991	128	RFC 1321
SHA-1 (Secure Hash Algorithm 1)	1995	160	FIPS PUB 180-1
SHA-2 (Secure Hash Algorithm Version 2)	2002	224, 256, 384, 512	FIPS PUB 180-2 FIPS PUB 180-3 FIPS PUB 180-4
SHA-3 (Secure Hash Algorithm Version 3)	2008	224, 256, 384, 512	FIPS PUB 202

ГОСТ Р 34.11-2012

Общая схема работы

- **Инициализационный вектор**:
  - вектор, определенный как начальная точка работы функции хэширования.
- **Функция сжатия**:
  - итеративно используемая функция, преобразующая строку бит длины L_1 и полученную на предыдущем шаге строку бит длины L_2 в строку бит длины L_2.



Основные обозначения

V^*	множество всех двоичных векторов-строк конечной размерности, включая пустую строку.
$\|A\|$	размерность (число компонент) конечного вектора A.
V_n	множество всех n-мерных двоичных векторов.
$\oplus$	операция покомпонентного сложения по модулю 2 двух двоичных векторов одинаковой размерности.
$A\|\|B$	конкатенация конечных векторов A и B.
A^n	конкатенация n экземпляров вектора A.
M	двоичный вектор, подлежащий хэшированию.
H	функция хэширования, отображающая двоичный вектор (сообщение) M в вектор (хэш-код) $H(M)$.
IV	инициализационный вектор функции хэширования.

Основные обозначения

$\mathbb{Z}_{2^n}$	кольцо классов вычетов по модулю 2^n.
$\boxplus$	операция сложения в кольце классов вычетов по модулю 2^n.
$\mathrm{Vec}_n\colon \mathbb{Z}_{2^n} \to V_n$	биективное отображение, ставящее в соответствие целому числу из кольца классов вычетов по модулю 2^n его двоичное представление.
$\mathrm{Int}_n\colon V_n \longrightarrow \mathbb{Z}_{2^n}$	отображение, обратное отображению Vec_n.
$\mathrm{MSB}_n\colon V^* \to V_n$	отображение, ставящее в соответствие вектору $z_{k-1}\|\|\ldots\|\|z_1\|\|z_0$, где $k \geq n$, $z_{k-1}\|\|\ldots\|\|z_{k-n+1}\|\|z_{k-n}$.
$\Phi\Psi$	произведение отображений, при котором отображение Ψ действует первым.

Инициализационные векторы

- Длина хэш-значения 256 бит:
  - $IV = 0^{512}$.
- Длина хэш-значения 512 бит:
  - $IV = (00000001)^{64}$.

Преобразования байтов и векторов в составе блока данных

- Нелинейное биективное преобразование $\pi = Vec_8\,\pi'\,Int_8 : V_8 \to V_8$:
  - замена байт;
  - $\pi' = \begin{pmatrix} 0 & 1 & \dots & 254 & 255 \\ 252 & 238 & \dots & 99 & 182 \end{pmatrix}$.
- Перестановка байт $\tau \in S_{64}$:
  - $\tau = \begin{pmatrix} 0 & 1 & \dots & 62 & 63 \\ 0 & 8 & \dots & 55 & 63 \end{pmatrix}$.
- Линейное преобразование множества двоичных векторов l:
  - набор 64-разрядных двоичных векторов b_i преобразуется путем умножения каждого вектора на двоичную матрицу $\mathbf{A}$ над полем F_2;
  - $l: c_i = b_i \cdot \mathbf{A}$, где $b_i, c_i \in V_{64}$.

Преобразования блока данных

- Состояние блока данных:
  - $a \in V_{512}$.
- Поразрядное сложение по модулю 2 векторов $a, k \in V_{512}$:
  - $X[k](a) = k \oplus a$;
- Замена байт:
  - $S(a) = S(a_{63} \| \cdots \| a_0) = \pi(a_{63}) \| \cdots \| \pi(a_0)$;
- Перестановка байт:
  - $P(a) = P(a_{63} \| \cdots \| a_0) = a_{\tau(63)} \| \cdots \| a_{\tau(0)}$;
- Замена 64-разрядных слов через матричное умножение над полем F_2:
  - $L(a) = L(a_7 \| \cdots \| a_0) = l(a_7) \| \cdots \| l(a_0)$.

Функция сжатия

- Функция сжатия $g_N: V_{512} \times V_{512} \to V_{512}, N \in V_{512}$ добавляет очередной блок сообщения m_i к результату преобразования предыдущих блоков сообщения $m_1, m_2, \dots, m_{i-1}$:
  - $g_N(h, m) = E(LPS(h \oplus N), m) \oplus h \oplus m$;
  - $E(K, m) = X[K_{13}]LPSX[K_{12}] \dots LPSX[K_2]LPSX[K_1](m)$;
  - $K_1 = K$;
  - $K_i = LPS(K_{i-1} \oplus C_{i-1}), i = 2, \dots, 13$;
  - C_{i-1} – итерационные константы, определенные в стандарте.

Процедура вычисления хэш-функции

- **Этап 1.**
  - 1.1. $h := IV;$
  - 1.2. $N := 0^{512} \in V_{512};$
  - 1.3. $\Sigma := 0^{512} \in V_{512};$
  - 1.4. Перейти к этапу 2.

- **Этап 2.**
  - 2.1. Проверить условие $|M| < 512$. При положительном исходе перейти к этапу 3. В противном случае выполнить последовательность вычислений:
  - 2.2. Вычислить подвектор $m \in V_{512}$ сообщения M: $M = \acute{M} \parallel m;$
  - 2.3. $h := g_N(h, m);$
  - 2.4. $N := \mathrm{Vec}_{512}(\mathrm{Int}_{512}(N) \boxplus 512);$
  - 2.5. $\Sigma := \mathrm{Vec}_{512}\left(\mathrm{Int}_{512}(\Sigma) \boxplus \mathrm{Int}_{512}(m)\right);$
  - 2.6. $M := \acute{M};$
  - 2.7. Перейти к шагу 2.1.

Процедура вычисления хэш-функции

- **Этап 3.**

3.1. $m := 0^{511-|M|} \| 1 \| M;$

3.2. $h := g_N(h, m);$

3.3. $N := \mathrm{Vec}_{512}(\mathrm{Int}_{512}(N) \boxplus |M|);$

3.4. $\Sigma := \mathrm{Vec}_{512}\big(\mathrm{Int}_{512}(\Sigma) \boxplus \mathrm{Int}_{512}(m)\big);$

3.5. $h := g_0(h, N);$

3.6. $h := \begin{cases} g_0(h, \Sigma), & \text{для хэш} - \text{кода длиной } 512 \text{ бит;} \\ \mathrm{MSB}_{256}\big(g_0(h, \Sigma)\big), & \text{для хэш} - \text{кода длиной } 256 \text{ бит;} \end{cases}$

3.7. Конец работы алгоритма.

Спасибо за внимание!

Евсютин Олег Олегович
Заведующий кафедрой информационной безопасности киберфизических систем
Канд. техн. наук, доцент

+7 923 403 09 21
oevsyutin@hse.ru