

Ежегодная международная научно-практическая конференция
«РусКрипто'2022»

**О реализации хэш-функции ГОСТ 34.11-2018
в виде квантовой схемы**

Денисенко Д.В., Рудской В.И.
МГТУ им. Баумана, ТК 26

Оценки для SHA-2, SHA-3, SM3

- Согласно [1,2] для реализации SHA-2, SHA-3, SM3 требуется 2402, 3200 и 2721 логических кубит.
- Согласно [3] для реализации SHA-2 с длиной хэш-кода 256 бит требуется 802 логических кубит.

1. Amy M., Matteo O., Gheorghiu, V., Mosca M., Parent A. Schanck, J. Estimating the cost of generic quantum pre-image attacks on SHA-2 and SHA-3, 2016, <http://arxiv.org/abs/1603.09383>.
2. Song G., Jang K., Kim H. Grover on SM3, 2021, <http://eprint.iacr.org/2021/668>.
3. Kim P., Han D., Jeong K.C. Time–space complexity of quantum search algorithms in symmetric cryptanalysis: applying to AES and SHA-2, Quantum Inf Process, 2018, <http://arxiv.org/abs/1805.05534>.

Оценки для ГОСТ 34.11-2018?

Итерационная конструкция Меркля-Дамгорда:

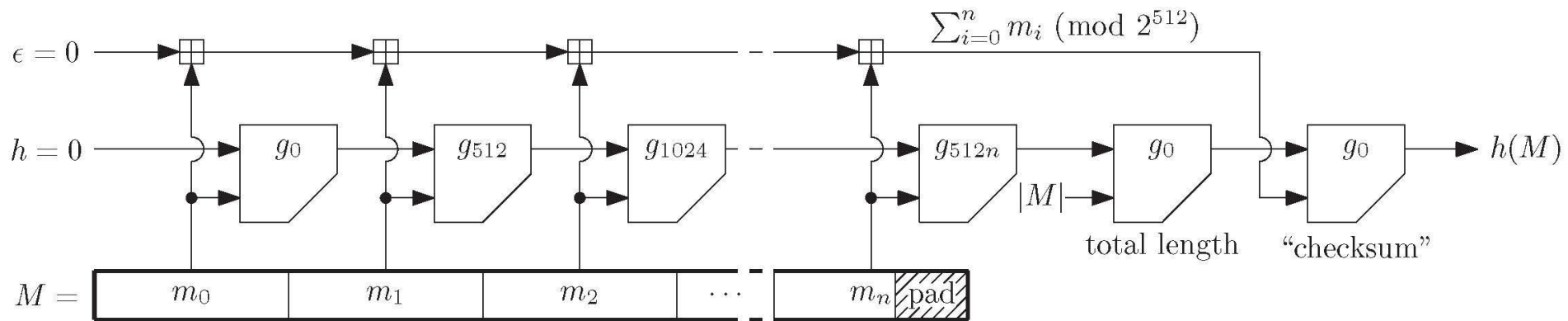
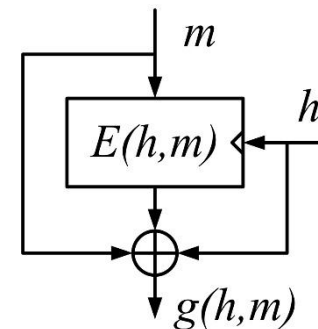


Схема из работы Markku-Juhani O. Saarinen «STRIBOB: Authenticated Encryption from GOST R 34.11-2012 LPS Permutation», eprint: 2014-271

Функция сжатия основана на конструкции Миягучи-Принеля:

$$g(h, m) = E(h, m) \oplus h \oplus m$$



ГОСТ 34.11-2018

Значение хэш-кода сообщения $M \in V^*$ вычисляется с использованием итерационной процедуры.

При этом на каждой итерации вычисления хэш-кода используется функция сжатия:

$$g_N: V_{512} \times V_{512} \rightarrow V_{512}, \quad N \in V_{512},$$

$$g_N(h, m) = E(LPS(h \oplus N), m) \oplus h \oplus m, \text{ где}$$

$$E(K, m) = X[K_{13}]LPSX[K_{12}] \dots LPSX[K_2]LPSX[K_1](m),$$

вычисление значений $K_i \in V_{512}$, $i = 1, \dots, 13$, осуществляется следующим образом:

$$K_1 = K;$$

$$K_i = LPS(K_{i-1} \oplus C_{i-1}), \quad i = 2, \dots, 13.$$

Модель ГОСТ 34.11-2018

Значение хэш-кода сообщения $M \in V^*$ вычисляется с использованием итерационной процедуры. При этом на каждой итерации вычисления хэш-кода используется функция сжатия:

$$g_N: V_5 \times V_5 \rightarrow V_5, \quad N \in V_5,$$

$$g_N(h, m) = E(S(h \oplus N), m) \oplus h \oplus m,$$

$$E(K, m) = X[K_{13}]SX[K_{12}] \dots SX[K_2]SX[K_1](m),$$

вычисление значений $K_i \in V_5, i = 1, \dots, 13$, осуществляется следующим образом:

$$\begin{aligned} K_1 &= K; \\ K_i &= S(K_{i-1} \oplus C_{i-1}), \quad i = 2, \dots, 13. \end{aligned}$$

$S = \{19, 11, 20, 28, 7, 10, 3, 29, 30, 9, 31, 1, 4, 25, 21, 17, 14, 5, 18, 2, 12, 22, 16, 23, 27, 0, 26, 15, 6, 8, 24, 13\}$ – случайно выбранная подстановка, $S: V_5 \rightarrow V_5$.

Константы $C_i \in V_5, (i = 1, 2, \dots, 12)$ также выбраны случайно: $\{18, 9, 14, 7, 8, 17, 6, 18, 13, 22, 12, 7\}$

Модель ГОСТ 34.11-2018

- Алгоритм вычисления хэш-кода $H(M)$ рассматриваемой модели совпадает с алгоритмом вычисления хэш-кода в соответствии с ГОСТ 34.11-2018 (за исключением длины двоичных векторов, вместо 512 бит используются 5 бит, $IV=00000$).

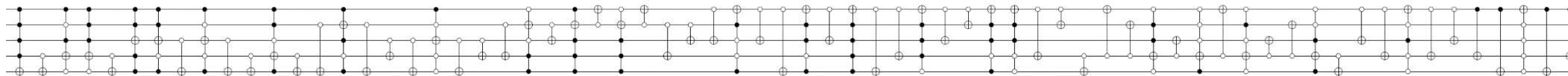
Рассматривались два случая:

- Если $|M| < 5$, то для вычисления $H(M)$ требуется вычислить g **три раза**
(см. ГОСТ 34.11-2018, п. 3.2, 3.5, 3.6 в описании процедуры вычисления хэш-функции)
- Если $5 \leq |M| < 10$, то для вычисления $H(M)$ требуется вычислить g **четыре раза**
(см. ГОСТ 34.11-2018, п. 2.3, 3.2, 3.5, 3.6 в описании процедуры вычисления хэш-функции)

Контрольные примеры для представленной модели ГОСТ 34.11-2018

Сообщение M ->	H(M)	Сообщение M ->	H(M)	Сообщение M ->	H(M)
{0,0,0,0}	{0,1,0,0,1}	{0,0,0,0,0}	{1,0,1,0,0}	{1,0,0,0,0}	{0,1,0,1,0}
{0,0,0,1}	{1,0,0,0,1}	{0,0,0,0,1}	{1,0,1,1,1}	{1,0,0,0,1}	{1,0,0,1,1}
{0,0,1,0}	{1,0,0,0,1}	{0,0,0,1,0}	{0,0,1,1,1}	{1,0,0,1,0}	{0,0,1,0,0}
{0,0,1,1}	{0,0,0,0,1}	{0,0,0,1,1}	{1,0,1,1,1}	{1,0,0,1,1}	{0,1,1,1,0}
{0,1,0,0}	{1,0,1,1,0}	{0,0,1,0,0}	{1,1,0,1,1}	{1,0,1,0,0}	{0,1,1,1,0}
{0,1,0,1}	{0,1,0,1,0}	{0,0,1,0,1}	{1,0,0,0,1}	{1,0,1,0,1}	{0,1,1,0,0}
{0,1,1,0}	{1,1,0,1,0}	{0,0,1,1,0}	{1,0,1,1,1}	{1,0,1,1,0}	{0,1,0,0,0}
{0,1,1,1}	{0,1,1,1,0}	{0,0,1,1,1}	{0,0,1,0,0}	{1,0,1,1,1}	{1,1,1,0,0}
{1,0,0,0}	{0,1,1,1,0}	{0,1,0,0,0}	{0,0,1,0,0}	{1,1,0,0,0}	{1,1,0,1,0}
{1,0,0,1}	{1,1,1,1,1}	{0,1,0,0,1}	{0,1,0,1,1}	{1,1,0,0,1}	{1,1,1,1,0}
{1,0,1,0}	{1,0,0,0,0}	{0,1,0,1,0}	{0,1,1,1,0}	{1,1,0,1,0}	{0,0,1,1,1}
{1,0,1,1}	{0,1,0,1,0}	{0,1,0,1,1}	{1,0,0,0,0}	{1,1,0,1,1}	{0,1,1,0,0}
{1,1,0,0}	{0,1,1,0,0}	{0,1,1,0,0}	{1,0,1,0,0}	{1,1,1,0,0}	{0,1,1,1,0}
{1,1,0,1}	{0,0,0,1,0}	{0,1,1,0,1}	{1,0,0,0,0}	{1,1,1,0,1}	{1,0,1,1,1}
{1,1,1,0}	{0,1,1,1,0}	{0,1,1,1,0}	{1,1,1,1,0}	{1,1,1,1,0}	{0,0,0,0,0}
{1,1,1,1}	{0,0,1,1,0}	{0,1,1,1,1}	{0,1,0,0,0}	{1,1,1,1,1}	{1,1,1,1,0}

Представленная модель хэш-функции ГОСТ 34.11-2018 реализована в квантовом симуляторе Quipper



Квантовая схема, реализующая
 $S = \{19, 11, 20, 28, 7, 10, 3, 29, 30, 9, 31, 1, 4, 25, 21, 17, 14, 5, 18, 2, 12, 22, 16, 23, 27, 0, 26, 15, 6, 8, 24, 13\}$
(всего 68 квантовых операторов)

1. Сумма по модулю 2^5 реализована с помощью сумматора из работы Takahashi Y. et al. «Quantum Addition Circuits and Unbounded Fan-out», 2009, arXiv:0910.2530v1, в котором 1 вспомогательный кубит можно опустить (т.е. для реализации суммы двух n -битных чисел по модулю 2^n достаточно $2n$ кубитов).
2. Функция сжатия g реализована на трех регистрах по 5 кубит каждый (всего: $message, h, m, n$ – для $|message|$ потребуется доп. регистр m , в n раскручиваем ключи).
3. В случае когда $5 \leq |M| < 10$ для вычисления $H(M)$ требуется дополнительная процедура для перевода кубитов в первоначальное «нулевое» состояние, соответствующая вычислению $g^{-1}_0(IV, m)$ при известном m .

**Квантовые ресурсы для формирования
(M,H(M))
при |M| = 4,
кол-во кубит при |M| <5**

Результат вызова функции GateCount
в квантовом симуляторе Quipper:

4: "H, arity 1"
19: "Init0"
1: "Init1"
172: "X, arity 1"
6000: "not, arity 1" controls 0+1
150: "not, arity 1" controls 0+4
1043: "not, arity 1", controls 1
600: "not, arity 1" controls 1+3
1050: "not, arity 1" controls 2+2
1200: "not, arity 1" controls 3+1
600: "not, arity 1", controls 4

Total gates: 10839
Inputs: 0
Outputs: 20
Qubits in circuit: 20

**Квантовые ресурсы для формирования
(M,H(M))
при |M| = 6,
кол-во кубит при 5 ≤ |M| <10**

Результат вызова функции GateCount
в квантовом симуляторе Quipper:

6: "H, arity 1"
24: "Init0"
1: "Init1"
288: "X, arity 1"
9520: "not, arity 1" controls 0+1
238: "not, arity 1" controls 0+4
1665: "not, arity 1", controls 1
952: "not, arity 1" controls 1+3
16: "not, arity 1", controls 2
1666: "not, arity 1" controls 2+2
1904: "not, arity 1" controls 3+1
952: "not, arity 1", controls 4

Total gates: 17232
Inputs: 0
Outputs: 25
Qubits in circuit: 25

Оценки минимального количества кубитов для реализации ГОСТ 34.11-2018

Если вместо 5 кубитных квантовых регистров рассматривать 512 кубитные квантовые регистры, то получим квантовые схемы, реализующие хэш-функцию ГОСТ 34.11-2018.

- Если $|M| < 512$, выполняются 3 вызова функции сжатия g , то для формирования квантового регистра с $(M, H(M))$ требуется не менее $|M| + 3 \cdot 512$ кубит, т.е. не менее $|M| + 1536$ кубит.
- Если $512 \leq |M| < 1024$, выполняются 4 вызова функции сжатия g , то для формирования квантового регистра с $(M, H(M))$ требуется не менее $|M| + 3 \cdot 512$ кубит, т.е. не менее $|M| + 1536$ кубит.

Вопросы

???