ОБОЗРЕНИЕ
ПРИКЛАДНОЙ И ПРОМЫШЛЕННОЙ
Том 25 МАТЕМАТИКИ Выпуск 3
2018

**Д. С. Богданов**, **Ф. А. Дали**, **В. О. Миронкин** (Москва, лаб. ТВП; ТК 26; МИЭМ НИУ ВШЭ)**. Сравнительный анализ FT-режима с некоторыми древовидными режимами выработки хэш-кода.**

**Введение.** Разработка и исследование режимов распараллеливания работы хэш-функций, удовлетворяющих современным криптографическим требованиям, приобретает все большую актуальность в силу их активного применения в различных практических приложениях (проверка цифровой подписи, проверка целостности при передаче больших объемов данных, построение датчиков псевдослучайных последовательностей и т. д.).

Настоящая работа продолжает исследования, начатые в [1–4] и посвящена сравнительному анализу универсального древовидного режима выработки хэш-кода — «FT-режима» («**F**ormatted **T**ree mode»), разработанного Д. С. Богдановым, Ф. А. Дали, В. О. Миронкиным, и некоторых существующих на сегодняшний день режимов [5–9] описанных в [10].

Все расчеты проведены с помощью вычислительной системы со следующими техническими характеристиками: процессор Intel Core i3 7100U: 2400 МГц (2 ядра), видеокарта NVIDIA GeForce 940MX: 2 Гб, операционная система Windows 10 Home, оперативная память DDR4: 4 Гб. Программная часть написана на PyCharm (Community Edition) 2017.2.4.

**Результаты экспериментов.** Поскольку FT-режим является универсальным режимом по отношению к используемому базовому механизму выработки узлов дерева хэширования (внутренней функции), в настоящей работе были рассмотрены реализации FT-режима, использующие в качестве внутренней функции все функции сжатия, описанные в [10], а именно, RC2, Threefish и Keccak [11], в том числе и функцию сжатия, построенную на основе алгоритма блочного шифрования AES. В дополнение к указанному списку был рассмотрен и российский стандарт функции хэширования ГОСТ Р 34.11-2012 («Стрибог») [12].

Для дальнейшего изложения введем ряд используемых определений и обозначений.

О п р е д е л е н и е 1. Под вычислительной трудоемкостью $T$ режима выработки хэш-кода будем понимать общее число элементарных операций (количество вычислений значений внутренней функции $h(x)$, $x \in V_m$), необходимых для выработки хэш-кода.

О п р е д е л е н и е 2. Под временной трудоемкостью $T^*$ режима выработки хэш-кода будем понимать общее время, затраченное на выработку хэш-кода.

З а м е ч а н и е 1. Результаты проведенных экспериментов касаются исключительно оценки характеристики $T^*$, поскольку вычислительная трудоемкость $T$ не зависит от используемой внутренней функции (измеряется в элементарных операциях) и принимает в лучшем случае значения порядка $O(\log n)$ (для режимов Skein, Sakura и FT-режима).

Так, в табл. 1 представлены приближенные значения временной трудоемкости $T^*$ выработки хэш-кода в FT-режиме для различных внутренних функций при фиксированных значениях длины сообщения и оптимальном подборе значений параметра арности, гарантирующих минимальные временные затраты.

**Таблица 1.** Значения $T^*$ для FT-режима

| **Длина сообщения, байты** | **AES** | **RC2** | **Threefish** | **Keccak** | **Стрибог** |
|---|---|---|---|---|---|
| $2^{21}$ | 0.2133 | 7.6547 | 1.6587 | 0.0902 | 16.1205 |
| $2^{22}$ | 0.4393 | 11.5778 | 3.4588 | 0.2745 | 30.9774 |
| $2^{24}$ | 3.0517 | 22.9384 | 12.3012 | 0.4554 | 181.8706 |
| $2^{25}$ | 6.0112 | 68.6577 | 25.8841 | 0.7819 | 247.0934 |
| $2^{27}$ | 8.9500 | 171.0502 | 44.5411 | 1.2068 | 325.5004 |

Как видно из табл. 1, максимальное быстродействие FT-режима достигается при использовании внутренней функции Keccak, поэтому в дальнейшем будем рассматривать FT-режим именно с указанной внутренней функцией.

На рис. 1 изображена зависимость $T^*$ от длины исходного сообщения при оптимальном выборе значений параметра арности (в зависимости от величины $n$) для указанных внутренних функций FT-режима.

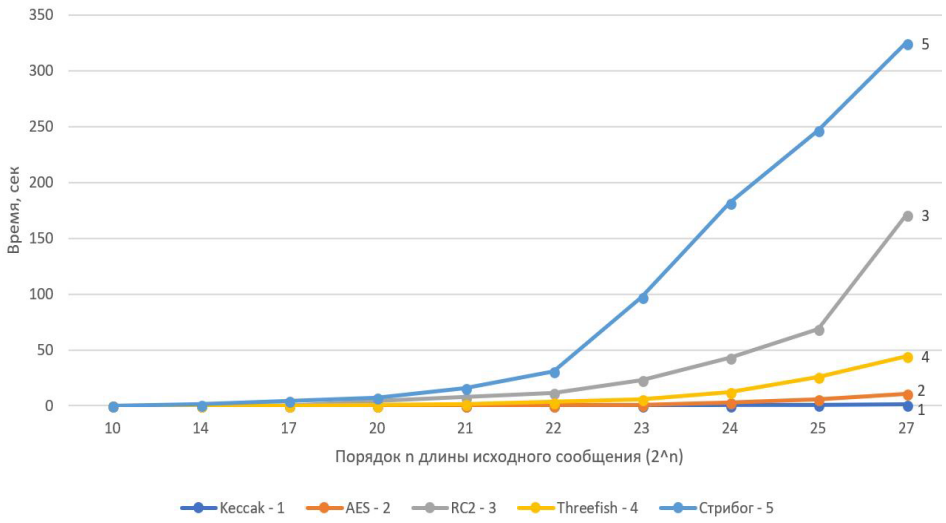


Рис. 1. Поведение $T^*(n)$ для различных внутренних функций в FT-режиме

З а м е ч а н и е 2. Для сравнения эффективности использования внутренних функций в качестве контрольных примеров были рассмотрены сообщения длины вида $2^k$, $2^k \pm 1$, а также $4^k$ и $4^k \pm 1$, $k \in \mathbb{N}$ (данные значения являются наиболее показательными для бинарного и 4-арного древовидных режимов Sakura и MD6 соответственно).

Результаты сравнения временной трудоемкости вычисления хэш-кода с использованием FT-режима и древовидных режимов [10] для оптимальных значений параметров хэширования представлены в табл. 2. Нетрудно заметить, что реализация FT-режима становится эффективнее по сравнению с другим режимами при хэшировании сообщений большой длины (наименьшие значения характеристики $T^*$ выделены жирным шрифтом).

**Таблица 2.** Некоторые значения $T^*$ для различных древовидных режимов

| **Длина сообщения, байты** | **Sakura** | **MD6** | **BLAKE2** | **Phash** | **Skein** | **FT-режим (Keccak)** |
|---|---|---|---|---|---|---|
| $4^{12}-1$ | 7.9994 | 95.0976 | 73.4584 | 16.3489 | **0.3117** | 0.4550 |
| $4^{12}$ | 7.7262 | 95.0986 | 73.4590 | 16.3498 | **0.3125** | 0.4554 |
| $4^{12}+1$ | 7.8109 | 96.7101 | 73.4598 | 16.3505 | **0.3238** | 0.4561 |
| $2^{27}-1$ | 21.2774 | 149.2800 | 124.9943 | 24.5148 | **1.2039** | 1.2062 |
| $2^{27}$ | 20.8756 | 149.2811 | 124.9956 | 24.5154 | **1.2054** | 1.2068 |
| $2^{27}+1$ | 21.0063 | 149.2838 | 124.9971 | 24.5167 | 1.2079 | **1.2075** |
| $4^{17}-1$ | 85.6961 | 176.2039 | 140.1895 | 99.8965 | 7.0370 | **6.9984** |
| $4^{17}$ | 83.0345 | 176.2058 | 140.1931 | 99.8977 | 7.0394 | **6.9994** |
| $4^{17}+1$ | 84.1254 | 179.0152 | 140.1997 | 99.9154 | 7.0451 | **7.0017** |

На рис. 2 представлены графики поведения функционала $T^*(n)$ для соответствующих древовидных режимов.

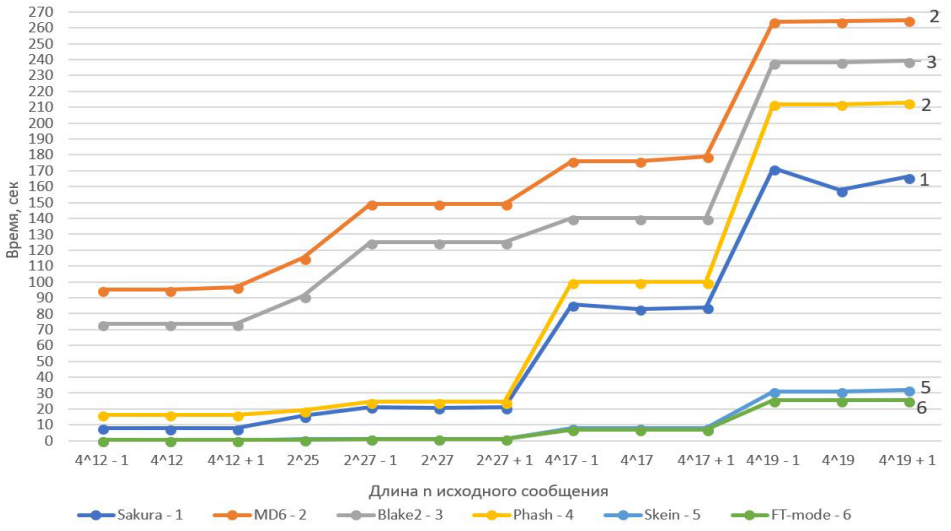


Рис. 2. Графики поведения $T^*(n)$ для различных древовидных режимов

**Заключение.** Экспериментальные исследования показали, что универсальный по отношению к механизму выработки узлов дерева хэширования FT-режим при использовании внутренней функции Keccak практически во всех случаях эффективнее с точки зрения вычислительной и временной трудоемкости большинства известных на сегодняшний момент древовидных режимов выработки хэш-кода. При этом для криптографического анализа FT-режим является наиболее «прозрачным», а его надежность полностью определяется криптографической стойкостью используемой внутренней функции.

## СПИСОК ЛИТЕРАТУРЫ

1. *Дали Ф. А., Миронкин В. О.* О вероятностных характеристиках одного класса моделей древовидного хэширования. — Обозрение прикл. и промышл. матем., 2016, т. 23, в. 4, с. 345–347.
2. *Дали Ф. А., Миронкин В. О.* О некоторых моделях древовидного хэширования. — Обозрение прикл. и промышл. матем., 2017, т. 24, в. 4, с. 241–244.

3. *Дали Ф. А.*, *Миронкин В. О.* О древовидных режимах работы хэш-функций. — Проблемы информационной безопасности. Компьютерные системы. 2018, №1, с. 113–121.
4. *Миронкин В. О.*, *Урусов М. И.* О коллизиях деревьев Меркла. — Обозрение прикл. и промышл. матем., 2018, т. 25, в. 1, с. 84–88.
5. *Aumasson J.-P.*, *Neves S.*, *Wilcox-O'Hearn Z.*, *Winnerlein C.* BLAKE2: simpler, smaller, fast as MD5, 2013, https://blake2.net.
6. *Bertoni G.*, *Daemen J.*, *Peeters M.*, *Van Assche G.* Sakura: a flexible coding for tree hashing, Cryptology ePrint Archive, Report 2013/231, 2013, http://eprint.iacr.org.
7. *Ferguson N.*, *Lucks S.*, *Schneier B.*, *Whiting D.*, *Bellare M.*, *Kohno T.*, *Callas J.* The Skein Hash Function Family, Version 1.3., October, 2010.
8. *Kaminsky A., P. Radziszowski S.* A case for a parallelizable hash. Rochester, NY: Department of Computer Science Rochester Institute of Technology, USA MILCOM, November, 2008.
9. *Rivest R. L.*, *Agre B.*, *Bailey D. V.*, *Crutchfield C.*, *Dodis Y.*, *Fleming K. E.*, *Khan A.*, *Krishnamurthy J.*, *Lin Y.*, *Reyzin L.*, *Shen E.*, *Sukha J.*, *Sutherland D.*, *Tromer E.*, *Yin Y. L.* The MD6 hash function. A proposal to NIST for SHA-3, Cambridge, MA: Massachusetts Institute of Technology, October, 2008. http://csrc.nist.gov.
10. *Дали Ф. А.*, *Миронкин В. О.* Обзор подходов к построению древовидных режимов работы некоторых хэш-функций. — Проблемы информационной безопасности. Компьютерные системы. 2017, №2, с. 46-55.
11. *Bertoni G.*, *Daemen J.*, *Peeters M.*, *Van Assche G.* The Keccak reference. http://noekeon.org.
12. *Kazymyrov A.*, *Kazymyrova V.* Algebraic Aspects of the Russian Hash Standard GOST R 34.11-2012, 2013. http://eprint.iacr.org/2013/556.pdf.