ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

***«*САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»**

**ИНСТИТУТ КОМПЬЮТЕРНЫХ НАУК И ТЕХНОЛОГИЙ**

**ВЫСШАЯ ШКОЛА ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ И СУПЕРКОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ**

**Отчет о прохождении   
стационарной производственной практики (научно-исследовательской работы на тему:**

«Сравнительное тестирование алгоритмов вычисления «быстрых» криптографических хэш-функций»**)**

Момота Даниэля Михайловича, гр. 3530903/60301

**Направление подготовки:** 09.03.03 Прикладная информатика

|  |
| --- |
| **Место прохождения практики:** СПбПУ, ИКНТ, ВШИСиСТ |

*(указывается наименование профильной организации или наименование структурного подразделения*

|  |
| --- |
|  |

*ФГАОУ ВО «СПбПУ», фактический адрес)*

**Сроки практики:** с 02.09.19 по 23.12.19

**Руководитель практики от ФГАОУ ВО «СПбПУ»:**

|  |
| --- |
| Кудрявцев Д.В., доцент ВШИСиСТ, к.т.н. |

*(Ф.И.О., уч. степень, должность)*

|  |
| --- |
| **Консультант от ФГАОУ ВО «СПбПУ»:**  Сергеев А.В., доцент ВШИСиСТ, к.т.н. |

*(Ф.И.О., должность)*

|  |
| --- |
|  |

**Оценка:**

Руководитель практики

от ФГАОУ ВО «СПбПУ» Кудрявцев Д.В.

Консультант

от ФГАОУ ВО «СПбПУ» Сергеев А.В.

Обучающийся Момот Д.М.

Дата:

Оглавление

[**ВВЕДЕНИЕ** 3](#_Toc27957514)

[1. Анализ предметной области и постановка задачи выпускной работы 4](#_Toc27957515)

[1.1. Хэш-функции 4](#_Toc27957516)

[1.2. Криптографические хэш-функции 5](#_Toc27957517)

[1.3. Классификация криптографических хэш-функций 6](#_Toc27957518)

[1.4. Быстрые КХФ, выбранные для тестирования 6](#_Toc27957519)

[1.5. Имеющиеся оценки скорости 7](#_Toc27957520)

[1.6. Постановка задачи, пути ее решения и планируемые результаты 8](#_Toc27957521)

[**ЗАКЛЮЧЕНИЕ** 10](#_Toc27957522)

[**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ** 11](#_Toc27957523)

# **ВВЕДЕНИЕ**

В настоящее время компьютерные сети, в том числе сеть Интернет, переживают период бурного развития. В связи с этим возрастает количество передаваемых по сети данных, в том числе конфиденциального характера. Поскольку зачастую передача данных ведется по открытым каналам, данные необходимо передавать в зашифрованном виде. Одним из способов шифрования являются криптографические хэш-функции.

Основными критериями оценки качества криптографических хэширующих алгоритмов являются скорость работы и криптостойкость. Обычно эти критерии противоречат друг другу, поэтому приходится искать компромиссные решения. Во многих случаях скорость является не менее важной, чем криптостойкость (при передаче значительных объемов данных, например в протоколах SSL, TLS и PPTP, в технологии VPN, а также при использовании криптопроцессоров; в облачных хранилищах, системах контроля версий).

Цель данной работы – сравнительное тестирование алгоритмов вычисления «быстрых» криптографических хэш-функций по критериям скорости работы и устойчивости к атакам методом грубой силы. Планируется сравнение алгоритмов MD5, SHA-1, BLAKE2, SHA-3, реализованных на языке C с использованием различных оптимизаций.

# Анализ предметной области и постановка задачи выпускной работы

В настоящее время важной частью взаимодействия по открытым сетевым каналам является шифрование данных. Предпочтительно, чтобы оно осуществлялось как можно быстрее, имело возможность распараллеливания, а также требовало относительно небольшого времени расшифровки. При этом криптостойкость примерно такой же приоритет, как и скорость.

Один из основных способов шифрования данных – криптографические хэш-функции (далее КХФ). Среди них могут быть выделены наиболее быстрые. Подробная характеристика хэш-функций, КХФ, а также четырех выбранных для реализации КХФ, приводится далее в этом разделе.

## Хэш-функции

Хэш-функции – это функции, предназначенные для «сжатия» произвольного сообщения или набора данных в некоторую битовую комбинацию фиксированной длины, называемую сверткой [1, стр. 347].

Введем более формальное определение хэш-функций. Пусть *X* – множество, элементы которого называются сообщениями (обычно – последовательности символов двоичного алфавита), *Y* – множество двоичных векторов фиксированной длины. Хэш-функцией называется всякая функция , легко вычислимая и такая, что для любого сообщения M значение (свертка) имеет фиксированную битовую длину [1, стр. 349].

В общем случае (согласно принципу Дирихле) одной свертке могут соответствовать более одного набора входных данных. Такая ситуация называется коллизией.

Смысл хэш-функций в том, что они позволяют «сжать» длинное сообщение в последовательность фиксированной длины, а с такими последовательностями гораздо проще оперировать. Это позволяет использовать их при построении ассоциативных массивов, поиске дубликатов в наборах «тяжелых» данных (например, изображений), вычислении контрольных сумм и других областях.

## Криптографические хэш-функции

Криптографическими хэш-функциями (далее КХФ) называются хэш-функции, пригодные для использования в криптографических системах. Для этого они должны удовлетворять следующим требованиям [2, стр. 82]:

1. Для любого значения сообщения *x* значение свертки *y=h(x)* должно вычисляться без значительных затрат времени и памяти.

2. Для любого значения *y* поиск *x*, такого что *y=h(x)*, должен требовать огромных затрат времени и/или памяти (стойкость к поиску первого прообраза).

3. Для любого значения *x* поиск *x’*, такого что и *h(x)=h(x’)*, должен требовать огромных затрат времени и/или памяти (стойкость к поиску второго прообраза).

4. Поиск *x* и *x’*, таких что и *h(x)=h(x’)*, должен требовать огромных затрат времени и/или памяти (стойкость к коллизиям).

Кроме того, важным свойством КХФ является лавинный эффект. Он состоит в том, что изменение значения даже одного бита шифруемого сообщения ведет к изменению большого количества бит шифра. В идеале меняются значения примерно половины битов. Если лавинный эффект для КХФ отсутствует или выражен слабо, надежность шифрования сильно понижается, т. к. по шифру можно (пусть с некоторой точностью) оценить значение шифруемых данных.

Еще одним важным свойством КХФ является высокая энтропия. В данном контексте слово «энтропия» означает, что значение хэш-функции выбирается из максимально широкого набора возможных вариантов. Это затрудняет атаку методом грубой силы.

## Классификация криптографических хэш-функций

КХФ можно разделить на два типа:

1. «Медленные» хэш-функции;

2. «Быстрые» хэш-функции.

«Медленные» хэш-функции (slow hash), также называемые функциями формирования ключа (Key Derivation Functions, KDF). Примеры: Bcrypt, SCrypt, PBKDF2, Argon2. Они применяются, в основном, для хэширования паролей и другой крайне ценной информации. Для них приоритетна криптостойкость, а не производительность. Более того, высокая скорость работы и небольшие затраты памяти для таких алгоритмов являются недостатком, т. к. облегчают атаку методом грубой силы [3, стр. 1].

«Быстрые» хэш-функции (fast hash). Такие функции, наоборот, создаются с расчетом на высокую эффективность. Они используются для подтверждения целостности данных, подтверждения подлинности источника данных, реализации цифровой подписи. В связи с этим, они должны демонстрировать высокую скорость работы, порядка пропускной способности среды передачи данных для возможности использования «в реальном времени» [4]. Примеры: MD5, SHA1, BLAKE2, семейство SHA2, SHA3 (Keccak), AES, DES.

## Быстрые КХФ, выбранные для тестирования

Для тестирования выбраны «быстрые» КХФ (потому что для «медленных» важной характеристикой является, скорее, «медленность» для всех архитектур). В настоящее время среди них («быстрых») наиболее популярными являются MD5, SHA1, функции семейства SHA2, SHA3 (Keccak) и BLAKE-2.

КХФ MD5 является одним из лидеров по производительности. Однако этот алгоритм определен как ненадежный (согласно [5], в нем могут быть найдены коллизии в течение одной минуты) и с 2011 года не рекомендуется к использованию в реальных системах [6].

SHA-1 тоже очень быстрый, однако Институт NIST (отвечает, в том числе, за стандарты ПО) планирует отказ от использования SHA-1 ввиду успешной атаки на него [7].

Семейство функций SHA-2, как и SHA-1 и MD-5, основано на использовании структуры Меркла-Дамгора. Так как SHA-1 и MD-5 оказались дискредитированы наличием у них уязвимостей, семейство SHA-2 определено как потенциально ненадежное (хотя успешных попыток взлома пока не представлено) [8]. Поэтому потребовалось проведение конкурса на создание нового алгоритма, основывающегося на другой идее (не на структуре Меркла-Дамгора).

Конкурс SHA-3 проводился в течение 2007 – 2012 годов. В финал вышли пять алгоритмов (BLAKE, Grøstl, JH, Keccak и Skein), имеющие различную внутреннюю структуру (это важно, так как взлом одного из них никак не поможет взлому других). Все они имеют достаточно высокую производительность и криптостойкость. Победителем вышел алгоритм Keccak, за счет использования нераспространенного метода шифрования – «функции губки», а также своей высокой производительности и нетребовательности к ресурсам, что позволяет запускать его даже на очень слабых устройствах, например USB-накопителях [9].

Алгоритм BLAKE2 является улучшенной версией BLAKE, авторы которого не успели его полностью доработать к конкурсу SHA-3. Он был представлен в конце 2012 года и является наиболее новым из описанных выше алгоритмов [10].

## Имеющиеся оценки скорости

По результатам нижеприведенных исследований, наиболее быстрыми из представленных выше алгоритмов являются MD-5, SHA-1 и BLAKE2.

По данным официального сайта BLAKE2 [10], он примерно равен по производительности SHA1 и превосходит по ней MD5.

Согласно докладу [11], BLAKE2 близок по скорости к MD5 и SHA-1.

Согласно данным исследования 2017 года [12], BLAKE2 быстрее MD5 в 1.5-2 раза, быстрее SHA-1 в 2-3 раза.

## Постановка задачи, пути ее решения и планируемые результаты

Как видно из представленного выше обзора имеющейся статистики, оценки производительности алгоритмов достаточно сильно различаются. В связи с этим, представляется полезным сравнить время работы этих алгоритмов «своими руками», чтобы получить подтверждение той или иной статистике. Также стоит сравнить устойчивость к атакам методом грубой силы с применением параллелизации вычислений (под мерой устойчивости понимается усредненное время, требуемое на взлом методом грубой силы).

Реализация алгоритмов вычисления криптографических хэш-функций будет производиться на языке С, как наиболее приближенном к ассемблерным инструкциям и в то же время достаточно легком для чтения и отладки. Планируется вначале написать «наивную» реализацию, затем попробовать улучшить ее с помощью использования таких техник, как оптимизация циклов, удаление излишних операций с памятью, раскрутка циклов, удаление зависимостей данных, реассоциация и др.

Кроме того, потребуется написать модули для точного определения времени работы хэш-функций и осуществления эффективной подстановки значений в рамках полного перебора.

Обработка результатов тестирования потребует применения методов математической статистики и обработки экспериментальных данных.

Результатом работы будет следующее:

1. Программный код реализации алгоритмов MD5, SHA-1, BLAKE2 и SHA-3 (Keccak);
2. Программная инфраструктура для тестирования скорости работы и времени взлома этих алгоритмов;
3. Результаты тестирования этих алгоритмов в виде сравнительных таблиц и графиков.

# **ЗАКЛЮЧЕНИЕ**

В данной работы была проведен анализ предметной области – криптографических хэш-функций (КХФ), где были рассмотрены их определение, классификация, наиболее популярные представители «быстрых» КХФ. На основе анализа предметной области были определены те из них, для которых сравнительное тестирование представляется наиболее важным.

Для определения наиболее производительных КХФ был проведен анализ имеющейся в открытом доступе статистики по скорости их работы. Был проведен анализ способов оптимизации программ на языке С, позволяющих добиться улучшения скорости работы.

На основе этих действий сформулирована цель исследования (сравнительное тестирование алгоритмов вычисления «быстрых» КХФ) и поставлены задачи: реализовать их на языке С, после чего сравнить их время работы и устойчивость к атаке методом грубой силы.

Таким образом, в результате данной работы были выбраны КХФ для сравнительного тестирования и определены инструменты и техники для их реализации.

# **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. А. П. Алфёров, А. Ю. Зубов, А. С. Кузьмин, А. В. Черёмушкин «Основы Криптографии», 2-е издание – Москва, 2002.
2. М. П. Минеев, В. Н. Чубариков «Лекции по арифметическим вопросам криптографии» – Москва, 2010.
3. Статья: Colin Percival / Stronger Key Derivation Via Sequential Memory-Hard Functions / BSDCan 2009 / Май 2009.
4. Differences Between Fast Hashes and Slow Hashes (эл. ресурс, время доступа 20.12.2019:

<https://openwall.info/wiki/john/essays/fast-and-slow-hashes>).

1. Vlastimil Klima / Tunnels in Hash Functions: MD5 Collisions Within a Minute / IACR Cryptology ePrint Archive / Апрель 2006.
2. S. Turner, L. Chen / Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms (RFC 6151) / Internet Engineering Task Force (IETF) / Март 2011.
3. <https://csrc.nist.gov/news/2006/nist-comments-on-cryptanalytic-attacks-on-sha-1>
4. <https://www.schneier.com/blog/archives/2005/11/nist_hash_works_4.html>
5. <https://www.nist.gov/news-events/news/2012/10/nist-selects-winner-secure-hash-algorithm-sha-3-competition>
6. <https://blake2.net/>
7. J-P. Aumasson, S. Neves, Z. Wilcox-O’Hearn, C. Winnerlein / BLAKE2: simpler, smaller, fast as MD5 / Доклад конференции ACNS 2013 / Июнь 2013.
8. <https://8th-dev.com/forum/index.php/topic,1200.msg6460.html>