Д.С. Мудров

# ИССЛЕДОВАНИЕ АЛГОРИТМОВ ХЭШИРОВАНИЯ ДЛЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ

(Самарский национальный исследовательский университет
имени академика С.П. Королева)

В современном мире, в котором у каждого есть мобильные устройства с доступом в интернет, еще более актуальной становится проблема защиты данных пользователей. Все больше программ и сервисов, дающих доступ к конфиденциальной информации, переносятся на мобильные устройства. К ним относятся «ГосУслуги», банковские приложения, мессенджеры, приложения для управления предприятиями и прочие сервисы, которые важны любому пользователю.

Злоумышленник, получивший доступ к аккаунту «жертвы», может распоряжаться его данными, деньгами, личной перепиской в своих корыстных целях. Это понесет негативные последствия для человека или даже предприятия, ставшего целью взлома [1].

Основным и самым действенным способом защиты пользователя является защита его аккаунта, путём хэширования пароля в клиентской части приложения, и передачи серверу хэш-результата, где уже сервер сверяет полученный хэш с хранящимся в базе данных. Ограничением в данном случае является малая производительность большинства мобильных устройств, в результате чего, длительная операция хэширования негативно сказывается на пользовательском опыте. Существует много видов хэш-функций, из которых самыми популярными и распространёнными являются:

- MD5;
- SHA-1;
- SHA256;
- SHA512;
- Bcrypt;
- Scrypt.

Bcrypt/Scrypt являются относительно новыми методами хэширования, главным преимуществом которых над остальными является отсутствие коллизий. Достигнуто это путем добавления случайной строки, так называемой криптографической соли, к исходным данным перед выполнением хэширования. Данный способ избавления от коллизий получил широкое применение, и используется вместе с другими алгоритмами хэширования, в которых данный метод не реализован изначально. Поэтому необходимо его реализовывать отдельно [2].

Данные алгоритмы хэширования были взяты из библиотеки Apache Commons и протестированы на мобильной операционной системе Android 8.0. Для алгоритмов MD5, SHA-1, SHA256, SHA512 перед хэшированием к исход-

ной строке добавлялась криптографическая соль. Так же для всех алгоритмов использовался метод растяжения, когда хэш-функция вычисляется рекурсивно несколько тысяч раз (рисунок 1).

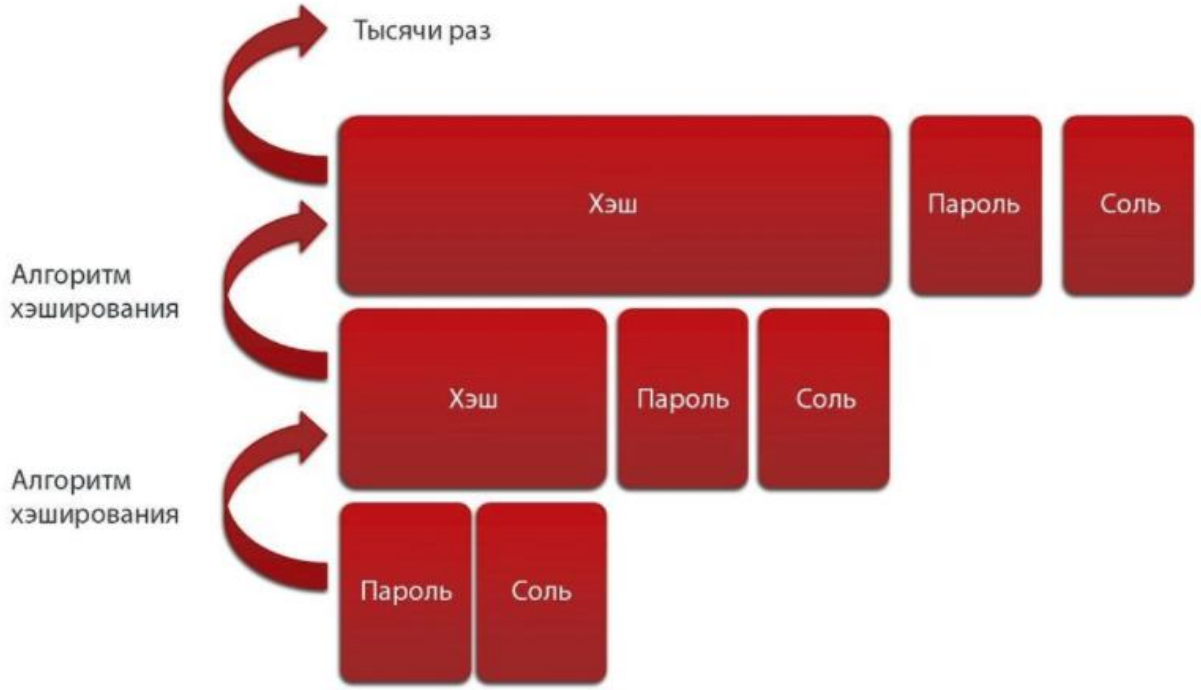


Рис. 1. Методы добавления криптографической соли и растяжения

Данный метод затрудняет взлом в десятки раз, для этого количество итераций должно быть таково, чтобы общее время вычислений заняло как минимум одну секунду. Чем более длительное получается хэширование, тем больше времени атакующему приходится тратить на взлом [3]. Однако, время авторизации не должно превышать 2 секунды, иначе это может сказаться негативно на пользовательском опыте. В таблице 1 представлены результаты работы данных алгоритмов хэширования.

Таблица 1. Время выполнения алгоритмов хэширования

| Название алгоритма: | Количество растяжений: | Время выполнения(сек.): | Количество растяжений: | Время выполнения(сек.): |
|---|---|---|---|---|
| MD5 | 1000 | 0.82 | 2000 | 2.15 |
| SHA-1 | 1000 | 0.95 | 2000 | 2.99 |
| SHA256 | 1000 | 1.59 | 2000 | 3.22 |
| SHA512 | 1000 | 1.71 | 2000 | 3.79 |
| Bcrypt | 1000 | 3.19 | 2000 | 6.91 |
| Scrypt | 1000 | 3.55 | 2000 | 7.43 |

Из полученных данных видно, что растяжение больше 2 тысяч раз нецелесообразно, потому что ни один из алгоритмов не уложился в 2-ух секундных рамках. Несмотря на свою высокую надежность и устойчивость к взлому,

bcrypt и scrypt при минимальном растяжении так же не уложились в заданный интервал. К тому же оба этих алгоритма создают сильную нагрузку на процессор, память и аккумулятор, что недопустимо при использовании на мобильных устройствах. Хэш-функции MD5 и SHA-1 не смогли выйти за рамки 1 секунды, что не соответствует требованиям. А также являются устаревшими и не рекомендуются для использования, так как уже не обеспечивают достаточно высокой надежности. Алгоритмы SHA256 и SHA512 исполняются в интервале от 1 до 2 секунд, поэтому являются оптимальным выбором. К тому же они генерируют более длинные хэши (в отличии от MD5 и SHA-1), что делает их более надежными и более устойчивыми к взлому.

**Литература**

1. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тесты на языке Си. – М.:Триумф, 2010. – 806 с.
2. Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В. Основы криптографии. – М.: Гелиос АРВ, 2001. — 407 с.
3. Блог компании Mail.Ru Group. Риски и проблемы хэширования паролей [Электронный ресурс]: – URL: https:habrahabr.ru/company/mailru/271245/ (дата обращения 20.03.2018).

Ф.М. Мухтаров

## МНОГОСТАДИЙНЫЕ ПРОЦЕССЫ ФОРМИРОВАНИЯ ИНФОРМАЦИОННЫХ РЕСУРСОВ

(Ташкентский университет информационных технологий
им. Мухаммад ал-Хоразмий. г. Ташкент)

**Аннотация:** *В данной статье приведены классификация информационные ресурсы, многостадийные процессы формирования информационные ресурсы, а также информатизация общества с учётом индустриального развития.*

**Ключевые слова:** информация, классификация, многостадийная, формирования, ресурс, информатизация, развития.

**Аннотация:** *Ушбу мақолада информацион ресурсларнинг классификациялари, информацион ресурслар шаклланишининг куп боскичли жараенлари, шунингдек, жамиятнинг ахборотлаштириш жараёнлари келтириб утилган.*

**Таянч сузлар:** информация, классификация, куп боскичли, шаклланиш, ресурс, ахборотлаштириш, ривожланиш.

**Abstract:** *Information resources, multistage processes of formation information resources and also informatization of society taking into account industrial development are given in this article classification.*