S. I. Todikov
Saint Petersburg Electrotechnical University «LETI»

MINIMISATION OF MULTIOPERATIONS IN A CLASS KEY STANDERD FORMS

Addresses the problem of minimizing multi-operations in a class of key standard forms. The work is based on the developed algorithm for minimizing multi-operations for n = 2 and n = 3 in the class of key standard forms. This algorithm is based on the analysis of multi-operations and the subsequent best replacements of all zero elements in order to obtain a minimal representation of multi-operations in the class of key standard forms. Using the developed algorithm, all minimal representations of multioperations for n = 2 and n = 3 in the class of key standard forms, the average complexity of the minimal representation of multioperations in the class of key standard forms and the quantitative distribution of multi-operations by the complexity of the obtained minimal representations are obtained. The obtained results of minimizing multi-operations in the class of key standard forms are compared with the minimization of multi-operations in the class of standard forms. When comparing the results, it is concluded that the minimal representation of multi-operations in the class of standard forms is better than in the class of key standard forms.

Multiprocess, superclones, key standard form, algorithm, minimization

УДК 004.056.53

Е. В. Шкляр, Е. Г. Воробьев, М. Ф. Савельев
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Распознавание клавиатурного почерка в браузере

Рассматривается тема распознавания клавиатурного почерка на современных мобильных и стационарных устройствах с установленными веб-браузерами. Выбор темы обусловлен массовым распространением Интернета в России и Мире – согласно исследованиям, к пользователям Интернета так или иначе относится половина населения планеты. Тем не менее, такой тип аутентификации практически не описан в существующих государственных стандартах РФ и мало исследован. В работе рассмотрены возможные методы решения задачи, возможные проблемы и ошибки, а также описан подход к разработке прототипа такой системы. Также приведен процесс разработки и схема работы прототипа системы распознавания клавиатурного почерка, работающая на устройствах любого типа. В работе предлагается инновационный метод аутентификации, не требующий от пользователя для входа в систему ничего, кроме ввода случайного текста. При успешной реализации этого метода станет возможной аутентификация на сайтах без логина и пароля.

Клавиатурный почерк, браузер, биометрия, аутентификация пользователя, мобильные устройства

Клавиатурный почерк – подвид поведенческой подгруппы аутентификации по неотчуждаемым признакам. Это довольно небольшая область, по всем аспектам и ответвлениям которой написано всего несколько десятков статей. В англоязычных источниках клавиатурный почерк называется «keystroke dynamics» [1]. Если переводить дословно, то это «динамика нажатия клавиш на клавиатуре».

В настоящей статье использована классификация методов аутентификации, предложенная исследователями С. Лью и М. А. Сильверман [2]. Они выделяют три типа аутентификации – по знанию, по владению и по неотчуждаемым признакам. К первому типу относятся пароли и PIN-коды, ко второму – смарт-карты, ключи и USB-токены, а третий признак охватывает широкую и популярную сейчас область биометрической аутентификации.

В наше время массово распространены и доступны мобильные устройства с дактилоскопическими датчиками или датчиками сетчатки глаза. Клавиатурный почерк представляет собой очень

узкий сегмент этой области – на него приходится менее 1 % всех устройств, в которые встроена биометрия. В государственном стандарте РФ «Биометрия» методы распознавания клавиатурного почерка и сопутствующие процедуры хранения биометрических шаблонов практически никак не раскрыты, что оставляет значительный простор для трактовки [3].

Распознавание клавиатурного почерка в браузере особенно актуально, поскольку браузерами пользуется больше половины населения планеты (4.021 млрд чел., исследование «Global Digital», 2018) [4]. Все они, так или иначе, взаимодействуют с клавиатурой – заходят на сайты, вводят пароли и т. д. В Интернет выходят с ПК 43 % пользователей, со смартфонов и планшетов – 56 %. В России же 75 % пользователей заходят в Интернет с ПК.

Деление механизмов биометрической аутентификации на две группы – поведенческую и физиологическую – использовано на основе [5].

Принцип анализа клавиатурного почерка. Клавиатурный почерк определяется по времени между нажатиями клавиш. При снятии биометрического шаблона клавиатурного почерка измеряют время нажатия двух, трех или четырех последовательных клавиш, сохраняют его, а на основе полученных значений строят математические модели для сравнения шаблонов нескольких пользователей [1]. Некоторые исследователи добавляют время нажатия одной конкретной клавиши, но эта метрика не показывает свою эффективность. Математическая модель получает на вход два биометрических шаблона – эталон и кандидата, а на выходе выдает рейтинг доверия – некую оценку схожести двух шаблонов. Чем выше рейтинг, тем больше вероятность, что пользователь, который пытается аутентифицироваться, не является злоумышленником.

Существует два вида распознавания клавиатурного почерка. Первый – распознавание на статическом тексте. Этим текстом может быть пароль или известная кодовая фраза. Здесь клавиатурный почерк служит вторым фактором аутентификации, а первый фактор – это пароль, который все-таки нужно знать злоумышленнику. Если он скомпрометирован, то отличия в биометрических шаблонах не позволят ему войти в систему.

Математически задача распознавания на фиксированном тексте довольно хорошо формализована. Нужно любым способом собрать информацию о времени между нажатиями соседних клавиш в тексте, сформировать из них вектор фиксированной размерности, который потом по кластерной модели (например, *k*-means в [6]) или любой другой модели сравнения векторов сравнивается с вектором-эталоном для этого же текста от этого же пользователя [7].

Второй тип – распознавание клавиатурного почерка при вводе псевдослучайного текста. Если пользовательский сеанс в информационной системе уже активен, то нет способа проверить личность пользователя, который проходил аутентификацию. Например, в офисе кто-то забыл заблокировать компьютер, а злоумышленник хочет этим воспользоваться. Тогда, если он попытается совершить противоправные действия, система обнаружения вторжений с настроенным анализом клавиатурного почерка сможет его вычислить и заблокировать компьютер. Это просто сделать, потому что сеанс открыт для конкретного пользователя, а его индивидуальный биометрический шаблон уже хранится в базе IDS.

В случае аутентификации по вводу случайного текста не существует надежных моделей формирования пользовательского шаблона и вычисления рейтинга. Здесь есть проблема в низких объемах выборки – злоумышленник во время работы в системе может напечатать два слова, по которым его будет сложно определить и отличить от легитимного пользователя. К тому же, время работы таких систем не позволяет в реальном времени оценить ситуацию и выдает результат через десятки минут, храня при этом много мегабайт векторов [8].

В нашем исследовании используется аутентификация на основе ввода случайного текста, который нужно брать из какого-то источника. Есть несколько вариантов – использование общедоступных генераторов текста на русском языке, фраз и предложений из русской литературы или же случайных слов из корпуса русского языка. В силу отсутствия хороших генераторов фраз в общем доступе и сложностью с обработкой русской литературы было принято решение использовать случайные слова.

Выделяют два основных численных показателя, которые определяют качество биометрической системы, – ошибки первого рода (FRR, количество ложноотрицательных) и второго рода (FAR, количество ложноположительных).

Ошибка первого рода (FRR – False Rejection Rate) – это вероятность ложного отказа в доступе. В нашем случае такая ошибка возникнет, если

человек, на клавиатурный почерк которого обучена система, не сможет пройти аутентификацию. Это может произойти из-за очевидных причин – повреждений одной или обеих рук, раннего времени доступа в систему (пользователь хочет спать), алкогольного опьянения и т. д; кроме того, возможны и не самые очевидные причины отказа в доступе – например, неверная работа решающего модуля, который и принимает решение о возможности доступа легитимного пользователя в систему. Практика показывает, что возможны небольшие отклонения от допустимых значений при распознавании пользователя.

Ошибка второго рода (FAR – False Acceptance Rate) – это вероятность ложного допуска, когда система ошибочно опознает чужого как своего. С этой характеристикой все проще – ни один «чужой» пользователь не должен смочь авторизоваться. Если в случае с аутентификацией по паролю злоумышленник, имея данные для входа, может легко получить доступ в систему, то в комплексе с модулем обработки клавиатурного почерка сделать это не представляется возможным. Кроме того, разрабатываемая программа не привязывается к конкретной парольной строке (хотя и имеет такую функциональность) и благодаря выбору случайных слов вместо пароля уменьшает вероятность ошибки второго рода.

Биометрические системы также иногда характеризуются коэффициентом равной вероятности ошибок 1-го и 2-го рода (EER – Equal Error Rates), представляющим точку совпадения вероятностей FRR и FAR. Надежная система должна иметь как можно более низкий уровень EER [9].

В общем случае формулы для вычисления FRR и FRP выглядят так:

FRR = [(Число попыток входа доверенного оператора) / (Число доступов в систему)] × 100 %;

FAR = (Число ложных срабатываний) / (Число ложных срабатываний + Число точных отказов).

У существующих систем показатели колеблются и зависят от метода. Например, у Ahmed и Traore FAR = 1.312 %, а FRR = 0.651 %. У Dowland, Furnell и J. D. Masters FAR = 0 % (что сложно реализуется и, скорее всего, есть следствие малой выборки), а FRR колеблется от 2 до 5 %. Это подтверждается и тем, что в исследовании J. D. Masters участвовало всего 10 человек, а у Dowland, Furnell не показаны результаты экспериментов так, чтобы по ним можно было оценить правдоподобность.

Технологически довольно сложно создать систему, которая в браузере будет анализировать данные в реальном времени и выдавать результат за секунды. В нашем исследовании предложен собственный метод сравнения биометрических шаблонов и разработан действующий прототип такой системы.

Дополнительная техническая сложность состоит в том, что система предусматривает работу в браузерах на смартфонах, поэтому нужно уметь обрабатывать нажатия на клавиши и на них. Эта задача пока полностью не решена, так как виртуальные клавиатуры на мобильных устройствах могут иметь разные методы ввода и, как показала практика, успешность обработки зависит от устройства, операционной системы, метода ввода и того, как именно человек пользуется клавиатурой на смартфоне. Например, в случае использования клавиатуры с непрерывным принципом ввода становится физически невозможно измерить время между нажатиями пар клавиш, потому что нажатий как таковых не происходит.

Схема работы системы распознавания в браузере. Для проверки гипотезы о возможности распознавания клавиатурного почерка в браузере был разработан прототип системы – веб-страница с текстовыми полями, возможностью загрузки текста с сервера и возможностью отправки данных о времени между нажатиями на клавиши. Схема представлена на рис. 1. Далее подробно раскроем механику работы и принципы действия систем принятия решений и коммуникации между сервером и клиентом.

Пользователь системы входит на сайт и в этот момент получает уникальный идентификатор. Единственной задачей скрипта session.php является сохранение информации о таймингах в файле текущей сессии и, если нужно, преобразование полученных кодов клавиш к единому формату для правильного хранения. Преобразование может понадобиться при запуске сайта на смартфонах под управлением iOS из-за особенностей обработки и нескольких устаревших методов фреймворка jQuery (например, метода KeyDown, который отвечает за обработку нажатия на клавишу).

Пользователь поочередно видит несколько случайных слов и вводит их по одному. Слова загружаются с сервера с помощью ajax-вызова файла word.php, единственная задача которого – выбор слова из словаря и отправка его на клиент. Сразу после окончания ввода слова появляется

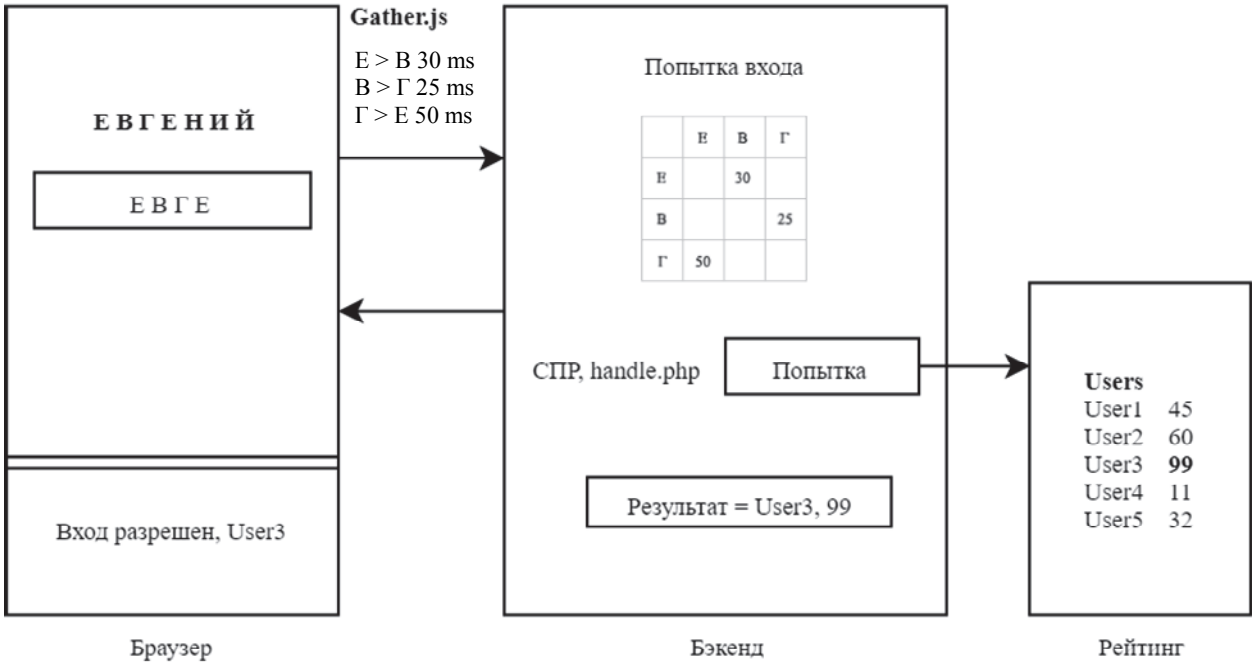

Рис. 1

```
$.ajax({
        type: "POST",
        url: "session.php",
        cache: false,
        data: {sessionID: sessionID, keycode: ecc, lastKeycode: lastKeycode, timing: timing},
        success: function(sesResponse) {

        },
        error: function(xhr, status, error) {
          console.log(xhr.responseText);
        }
                                    });}
```

Рис. 2

следующее – так происходит, пока все слова для данной сессии (обычно 5–7) не будут введены. Количество вводимых слов может варьироваться – их будет меньше, если общее количество введенных пар символов превысит 35 или если система принятия решения сделает однозначный вывод о том, кем является пользователь.

В фоновом режиме клиентский скрипт, работающий на JavaScript, измеряет время между нажатиями пар клавиш с помощью обработки кодов, полученных функцией KeyDown (рис. 3). На сервере формируется ($N \times N$)-матрица, где N – размер алфавита. В случае с русским языком матрица имеет размер 33 × 33, русский + английский алфавиты – 59 × 59. Номер столбца и номер строки – порядковый номер введенного символа в алфавите (А – 1, Я – 33), а в ячейку матрицы на пересечении строки и столбца записывается время ввода конкретной пары символов.

Схема ajax-запроса к файлу session.php приведена на рис. 2 – скриншоте окна браузера Chrome с запущенным сервисом.

К моменту окончания ввода последнего символа на сервере уже хранится полный файл для конкретной пользовательской сессии. Его особенность в том, что информация сразу записывается в файл с именем сессии, чтобы впоследствии легче обрабатываться системой принятия решения. Файл сессии состоит из наборов по три строки: две строки с кодами клавиш в формате Unicode и одна строка со временем в миллисекундах между их нажатием – и так для каждой пары символов.

В момент нажатия последней буквы сервер получает сообщение, что ввод завершен. В этот момент включается система принятия решения (СПР) в файле handle.php. Задача данного скрипта – это анализ введенного в рамках сессии шаблона и сравнение его со всеми шаблонами, которые уже зарегистрированы в базе, для определения максимально похожего и построения предположений о личности пользователя на их основе.

Математическая модель принятия решения в разработанном прототипе. Для удобства, как уже было упомянуто, все сессионные данные

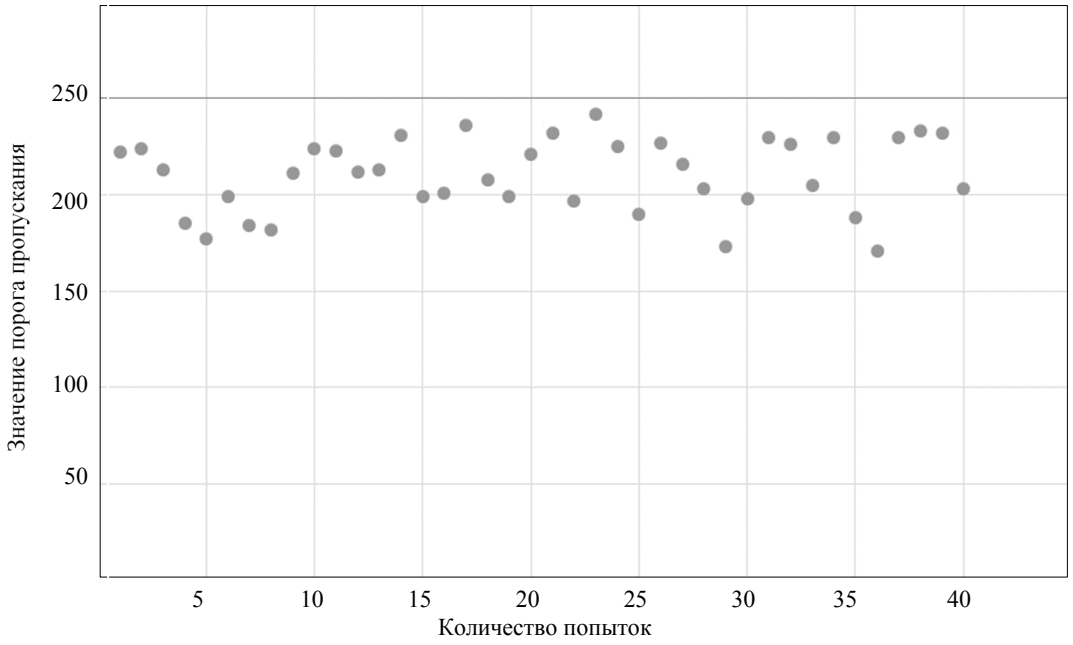

Рис. 3

переносятся в матрицу, размерность которой равна квадрату размерности алфавита, на котором работает система распознавания. Главной задачей служит определение меры схожести двух матриц – эталона и кандидата – и присвоение каждой паре определенного рейтинга, на основе которого можно отсортировать набор эталонов и выбрать самый подходящий.

Далее приведен алгоритм работы СПР:

1. СПР поэлементно проверяет каждую ячейку матрицы и подсчитывает количество ненулевых элементов, которые есть в соответствующих местах обеих матриц. Это количество записывается в переменную $couples.

2. Для каждой такой пары проводится проверка – если расхождение между временем нажатия у кандидата и эталона менее 20 %, переменная $s1 увеличивается на 1. Переменная $s2 увеличивается на 1 в любом случае.

3. В переменную $per суммируется значение времени нажатия из матрицы-кандидата.

4. Если на прошлых шагах переменные $s2 и $per не равны нулю, то нужно перейти к расчету рейтингов. Все четыре имеют значения от 1 до 100, где 1 – минимальное сходство с биометрическим шаблоном пользователя, а 100 – точное совпадение с этим шаблоном.

После того как для каждого пользователя, зарегистрированного в системе, вычислены рейтинги сравнения с кандидатом, проводится сортировка и определение того, может ли пользователь быть легитимным.

После сортировки всех успешных попыток выбирается та, у которой выше рейтинг R4, и на клиент возвращается сообщение об успешной авторизации определенного пользователя.

Эксперименты и результаты. С разработанной системой в настоящее время не проводилось массовых экспериментов для вычисления показателя ошибки второго рода. Одного из авторов система «узнаёт» в 80 % случаев. На рис. 3. приведен график одного из экспериментов, где обозначены 40 попыток вторжения в систему с одним шаблоном в базе. Самая частая проблема при единичных экспериментах – пользовательский рейтинг оказывается недостаточно высоким, чтобы определить схожесть с каким-либо из шаблонов. Теоретически, это должно решиться после анализа массовой аутентификации в сервисе и доработки порогов параметров аутентификации.

Конечно, у подобных систем аутентификации есть пространство для ошибок – любые предсказательные системы работают хуже полностью детерминированных, но это необходимое допущение для существования системы распознавания клавиатурного почерка на случайном тексте. Разработанный прототип потенциально может быть улучшен для снижения количества ошибок первого и второго рода следующим образом:

1. Слова для ввода будут выбираться с учетом фиксированного набора диграфов; случайно, но с применением методов машинного обучения с прямым распространением ошибки. Это позволит

охватывать наборы диграфов максимально полно, используя при этом разные наборы слов для всех пользователей.

2. В нынешней версии прототипа предусмотрена аутентификация без логина, с «угадыванием» его в процессе работы. Для стандартного входа на сайты можно добавить поле для ввода логина – оно, как предполагается, сильно увеличит успешность распознавания пользователей.

СПИСОК ЛИТЕРАТУРЫ

1. Axelsson, S. The base-rate fallacy and its implications for the difficulty of intrusion detection // Proc. of the 6th ACM conf. on Computer and communications security. New York, 1999. C 1–7.

2. Liu S., Silverman M. A. A. Practical guide to biometric security technology // IT Professional. 2001. Vol. 3. P. 27–32.

3. Информационные технологии. Биометрия. Обучающая программа по биометрии: ГОСТ Р 54412–2011. Введ. 2002-01-01. М.: Изд-во стандартов, 2001.

4. Digital in 2018: world's internet users pass the 4 billion mark. URL: https://wearesocial.com/blog/2018/01/global-digital-report-2018, свободный (дата доступа 12.12.2018).

5. Bergadano F., Gunetti D., Picardi C. User authentication through keystroke dynamic // ACM Transactions on Information and System Security (TISSEC). 2002. № 5 (4). P. 367–397.

6. Dowland P. S., Furnell S. M. A long-term trial of keystroke profiling using digraph, trigraph and keyword latencies. Security and protection in information processing System // IFIP 18th World Computer Congress TC11 19th Intern. Information Security Conf. Toulouse, 2004. P. 275–289.

7. Ahmed A. A. E., Traore I. Anomaly intrusion detection based on biometrics // Proc. from the Sixth Annual IEEE SMC. West Point, 2006. P. 452–453.

8. Lynch D. M. Securing against insider attacks // Information Security J.: A Global Perspective. 2006. Vol. 15, № 5. P. 39–47.

9. Technical document about FAR, FRR and EER. URL: http://syris.com, свободный (дата доступа 12.12.2018).

E. V. Shklyar, E. G. Vorobyev, M. F. Savelyev
Saint Petersburg Electrotechnical University «LETI»

BROWSER-BASED KEYSTROKE DYNAMICS RECOGNITION

This paper is up to detecting keystroke dynamics on the modern mobile devices and computers with web-browsers installed. Theme is based on mass distribution of the Internet in the Russia and in the other world. According to researches, a half of the world population is connected to the Internet. Anyway, this type of the authentication is not described well, it's also not represented in state standards. We consider possible approaches to solve this problem, possible issues, and also describe a scheme of the prototype. Also we described a full scheme of a tool for user authentication on any devices with a keyboard, neither physical or virtual. In a case of 100 % success of this method, it would be possible to log in without login and password – just after typing a few words.

Keystroke dynamics, browser, biometry, user authentication, mobile devices