Nexo Revista Científica
Vol. 34, No. 04, pp. 1258-1271/Septiembre 2021

ISSN-E 1995-9516
Universidad Nacional de Ingeniería

http://revistas.uni.edu.ni/index.php/Nexo
https://doi.org/10.5377/nexo.v34i04.12659

Developing of keystroke dynamics analyzing systems using a presentation based on state contexts

Desarrollo de sistemas de análisis de dinámica de pulsaciones mediante una presentación basada en contextos de estado

Dmitry A. Trokoz, Alexey I. Martyshkin*, Elena A. Balzannikova, Irina G. Sergina

Penza State Technological University, 440039, Russia, Penza, 1/11 Baydukova
proyezd/Gagarina ul., 1/11

alexey314@yandex.ru

(recibido/received: 19-mayo-2021; aceptado/accepted: 10-agosto-2021)

ABSTRACT

This aim of the article is to discusses the main static and dynamic user identification methods by keystroke dynamics. As part of the research, a generalized way of representing the process of typing on the keyboard based on the sequential change of the keyboard state was proposed. The definition of the keyboard state context, which is the basis for the dynamic identification procedure, is formulated. The proposed approach will make it possible to apply a variety of static identification methods, significantly expanding the set of methods used for dynamic user identification by keystroke dynamics.

Keywords: behavioral biometry, dynamics identification, keyboard, data analysis, machine learning.

RESUMEN

Este artículo analiza los principales métodos de identificación de usuarios estáticos y dinámicos por dinámica de pulsaciones de teclas. Como parte de la investigación, se propuso una forma generalizada de representar el proceso de escritura en el teclado basada en el cambio secuencial del estado del teclado. Se formula la definición del contexto del estado del teclado, que es la base del procedimiento de identificación dinámica. El enfoque propuesto permitirá aplicar una variedad de métodos de identificación estática, ampliando significativamente el conjunto de métodos utilizados para la identificación dinámica de usuarios mediante la dinámica de pulsaciones de teclas.

Palabras clave: biometría conductual, identificación dinámica, teclado, análisis de datos, aprendizaje automático.

1. INTRODUCTION

Today, the problems of preventing unauthorized access to confidential and personal information, its illegal distribution, as well as preventing illegal actions on behalf of another user are urgent tasks in the field of information security. Traditionally, cryptographic means and means of authentication based on knowledge (passphrase or answer to a specific question) or attribute (smart-card, key) are used to protect information systems. In addition, there is a class of biometric identification and authentication methods that are used both as stand-alone solutions and as a strengthening of traditional security tools. Biometric methods have a number of advantages over other methods based on knowledge or an attribute: the source of biometric data is always with the user, cannot be lost. Besides compromising and copying a biometric image is often extremely difficult. However, the main disadvantage of biometric methods can be attributed to changes in the initial biometric image of the user due to age-related changes, the psychophysical state of a person or the presence of injuries. This aspect may impede the identification procedure or make it completely impossible. In addition, biometric analysis often requires additional and often expensive equipment.

Among the biometric characteristics, static (fingerprint, iris or vein pattern) and behavioral (handwriting, gait, voice) can be distinguished. The analysis of static characteristics assumes single identification only at the moment of user login to the system. While the behavioral characteristics suitable for continuous identification procedure over a period of time. Many of them provide the ability to perform this procedure hidden from the user.

To perform both static and dynamic identification, standard PC input devices can be used: keyboard and mouse. This paper describes issue of dynamic user identification based on the unique characteristics of typing on the keyboard - keystroke dynamics. The main advantage of this approach, in addition to hidden and dynamic identification, is the absence of the need for additional equipment, since today almost every personal computer is equipped with this input device.

Static analysis of a user's keystroke dynamics assumes an identification procedure by entering a predetermined sequence of characters: a passphrase. A typical area of application of this type of analysis is to conduct a user authorization procedure when logging into the system using the user's login and password, which, in combination with the analysis of keystroke dynamics, increases the security of the system in the event of their compromise. To build an image of the keystroke dynamics for each user, several samples of entering a passphrase are enough, which, as a rule, does not require time.

Dynamic analysis, in turn, allows you to identify the user regardless of the text he is typing. This approach allows not only to discern the user during login, but also to perform continuous analysis of the keystroke dynamics throughout the entire session in order to recognize the change of the active user. In addition, this method makes it possible to execute the identification procedure not by a previously known password, but by entering a small piece of free text, thereby relieving the user and the system from the need to store passwords. This type of analysis requires a longer training period for the system to generate the most complete and accurate biometric image.

2. MATERIALD AND METHODS

Initially, methods of static analysis were developed in the field of keystroke dynamics analysis. In the 1980s, the National Science Foundation and the US National Bureau of Standards conducted a series of studies that concluded that typing patterns contain unique characteristics that can be identified. The researchers explained the psychological basis for using keystroke dynamics to give researchers a basic understanding of the various processes involved in typing process (Gaines et al., 1980). Extensive research shows that a person's typing is a behavioral characteristic that develops over a period of time and therefore cannot be shared, lost or forgotten. This parameter shows quite legible information that can be used for identification and authentication.

Today there are many classes of methods for static analysis of user keystroke dynamics. One of these classes is a variety of statistical methods, which are based on a statistical assessment of the main timing parameters of keyboard events: mean, median, standard deviation, probability density, and others. These methods were used both in early works on the study of keystroke dynamics, and in modern research.

So, in the first works in this area, works by R. Joyce and G. Gupta (1990), F. Monrose and A. Rubin (1997), D. Song, P. Venable, and A. Perrig (1997), methods based on statistical estimation of the distance between vectors of delays between keyboard events.

Statistical methods also include probabilistic methods, the main position of which is that the temporal characteristics of the keystroke dynamics correspond to the Gaussian distribution, which was considered in (Dowland, 2001). Also, this class of probabilistic statistical methods includes algorithms based on hidden Markov models (Ali et al., 2016), estimates of probability density, weighted probability (Bryukhomitsky, 2010).

Clustering methods can also be distinguished, which consist in combining similar vectors of characteristics into clusters. The purpose of algorithms of this type is to form clusters with the closest characteristics of the elements within the cluster, but the most different parameters of the elements between the clusters. Within the framework of the clustering method, the k-means method or methods of fuzzy sets are used (Mandujano & Soto, 2004). To estimate the distance between vectors in various studies, many estimates are used: Euclidean distance (Sidorkina & Savinov, 2013), Manhattan distance (Killourhy & Maxion, 2009), Mahalonobis distance (Killourhy & Maxion, 2008), a measure of disorder (Plank, 2016).

Another class of keystroke dynamics analysis methods are machine learning methods, which are often used in classification, pattern recognition, and clustering problems.

Among them, approach based on neural networks can be highlighted. A feature of the approach based on neural networks is the need for a large number of alien images to train the model. In addition, adding, deleting, or updating a biometric image in a database will generally lead to retraining of neural networks. To solve the problem of user identification by keystroke dynamics, various types of neural networks are used: direct propagation (Pavaday et al., 2007), multilayer neural networks (Pavaday & Soyjaudah, 2008), recurrent networks (Kobojek & Saeed, 2016) and self-organizing maps (Loy et al., 2007).

Decision trees also belong to the class of pattern recognition methods based on neural networks. In studies devoted to the analysis of keystroke dynamics, approaches based on different variants of the use of decision trees are used: a random forest and parallel decision trees (Sheng et al., 2005).

Also, the support vector machine learning approach should be noted. This method generates the smallest regions that cover the largest number of features of a certain class. This method maps an input vector into a multidimensional feature space using a kernel function (linear, polynomial, sigmoidal, or radial basis function). The algorithm will find a hyperplane that will divide the space of features in such a way that on one side there will be as many features of the image of "users" and as few features of "impostors" as possible, and correspondingly as many features of "impostors" and fewer of "users" the other side. As a result, the function will build a more complex decision boundary than linear methods. The use of the support vector machine in the problem of user identification by keystroke dynamics is considered in (Bhatia & Hanmandlu, 2017).

In the field of continuous identification, the methods used for static analysis, in the general case, cannot be applied. A different set of methods and algorithms are used to implement continuous analysis.

Research in the field of dynamic of keystroke dynamics analysis was first published in 1995 by Shepherd, who showed the possibility of user identification by keystroke dynamics in the process (Shepherd, 1995).

In addition, Dowland used digraph, trigraph, and word spacing in his study as a distance-based classifier and achieved an FAR of 4.9% and an FRR of 0% based on testing on 35 users (Dowland & Furnell, 2004).

In the works of Gunetti, a method for analyzing keystroke dynamics based on free text using a measure of randomness of intervals, presented in the form of digraphs, is considered. This approach made it possible to achieve 3.16% FAR and 0.02% FRR for 40 users classified as "user" and 165 users as "impostors" (Gunetti & Picardi, 2005).

In the field of dynamic analysis, there are several classes of methods that in many ways duplicate the methods for static analysis, but taking into account the peculiarities of the parameters of continuous analysis. Many statistical methods, machine learning methods and bioinspired algorithms are also used to solve this problem. In contrast to static analysis, in which the main parameter is a fixed-length feature vector, which depends on the analyzed characteristics and the length of the passphrase, in dynamic analysis, in most cases, digraphs, trigraphs, and other n-graphs consisting of 2, 3 or several feature sequences are used. Thus, in the case of dynamic user identification, it is not a fixed expression as a whole that is analyzed, but a sequence of n-graphs that are formed on the basis of an arbitrary text.

In addition, the study by Daniele Gunetti and Claudia Picardi (2005) examined the statistical characteristics of combinations of two, three or four consecutive characters, taking into account only specific combinations of characters for each individual user. A similar study is reflected in a series of articles by Fernel, in which the characteristics of two or three keys were analyzed in this way using not only statistical analysis, but also using neural networks (Dowland, 2001; Dowland et al., 2002; Dowland & Furnell, 2004).

So, in the work of Bours, Patrick and Mondal, Soumik (Bours & Mondal, 2015) for dynamic user recognition, a weighted estimate of the Euclidean distances between digraphs was used in combination with an ensemble of classifiers consisting of a feedforward neural network, a recurrent neural network and a support vector machine. A genetic algorithm was used to optimize the weights in the ensemble.

Thus, there is an extremely wide range of static identification methods that are used to analyze the temporal characteristics of the keystroke dynamics, presented mainly in the form of intervals between different types of events. These intervals can be calculated both between two successive events, such as, for example, the time between keystrokes or the time of holding the keys, and between a sequence of several successive events, forming digraphs or, in general, n-graphs. Let's designate this view model as an event model.

Dynamic analysis of keystroke dynamics is complicated by the fact that, unlike static identification, it is impossible to generate a feature vector of a fixed length. In this case, a continuous sequence of events and their time parameters are subjected to the identification procedure. This type of biometric identification has a slightly more limited set of methods and, as a rule, each of them develops its own way of representing the biometric image of the keystroke dynamics, which is most appropriate for a particular analysis method.
Based on this, a new way of representing the biometric image of the keystroke dynamics is proposed, which will be a more universal form of representing the biometric image, and which can serve as the basis for both static identification and dynamic methods. This model is based on the concept of the state of the keyboard. Let's call it a state-based model.

3. RESULTS AN DISCUSSION

Basic timing of keystroke dynamics

The data received from the keyboard is a sequence of events occurring at a specific time. Each event is a tuple of three parameters:

$$(\mathrm{Event_i}, \mathrm{Keycode_i}, \mathrm{Timestamp_i}), \forall \mathrm{i}, 0 \ \leq \mathrm{i} < \mathrm{n}$$
$$\mathrm{Keycode}_i \ \in \ \mathrm{Z}$$
$$\mathrm{Event_i} = \ \{P, R\}$$
$$\mathrm{Timestamp_i} \ \in N$$

Where i - is the index of the event, n - is the number of characters in the typed text.

The basic metrics for keyboard typing are the timestamps of two types of events: a key press and a key release. Let's denote ${P^{Keycode}}_{i}$(Timestamp) as the event of pressing the i key at time t and ${R^{Keycode}}_{i}$(Timestamp)as the event of releasing the i key at time t.

Based on these events, a number of features can be distinguished that can be the basis for the analysis of keystroke dynamics.

The primary features are the time intervals between a pair of events of different types. These intervals include: key hold time, delay, and time between keystrokes.
The dwell time of the i key can be defined as the interval between releasing and pressing the same key:

$$DT_i = R_i(t) - P_i(t) \tag{1}$$

Keystroke interval is defined as the interval between successive keystroke events (flight time):

$$FT_i = P_{i+1}(t) - P_i \tag{2}$$

In turn, define the delay as the interval between lowering the previous key and pressing the next one (latency time):

$$LT_i = P_{i+1}(t) - R_i(t) \tag{3}$$

If the key holding time is crossed, this sign will have a negative value. The described events and primary signs can be graphically represented as shown in Figure 1.

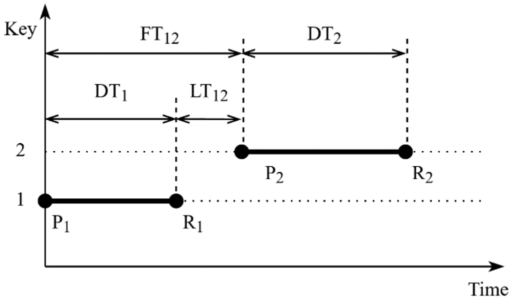

Figure 1. Primary features of keystroke dynamics

Based on the combination of the considered primary features, composite features can be formed: digraphs, trigraphs, etc. The specific type of the composite feature depends on the method used for analyzing the keystroke dynamics (Pashchenko et al., 2018).

Format representation of keyboard state

To form a generalized keyboard model, we represent it as a fixed set of keys, each of which at any moment of time can be in one of two states: pressed or released. Depending on the specific keyboard, the number of keys may vary. To build the model, we will restrict the set of keys only to those that, as a rule, are involved in typing. These keys include: character and number keys (47 keys of the main keyboard), as well as service keys: Space, Enter, Right Ctrl, Left Ctrl, Right

Shift, Left Shift, Right Alt, Left Alt, Caps Lock, Tab, BackSpace. Thus, the keyboard model will contain 58 keys, on the basis of which the timing characteristics will be estimated.
Let's imagine the state of the keyboard as a vector, each element of which contains the state of an individual key. The set of states, based on the proposed model, consists of two elements:

$$State = \{Pressed, Released\}$$

Therefore, the keyboard state vector Keyboard will look like this:

$$Keyboard = [State_1, State_2, \dots, State_i, \dots, State_{58}]$$

where i is the key number.

Since the proposed model has 58 elements, each of which can be in one of two states, the total number of possible states of the model is 2 ^ 58, which is more than a quadrillion states. Obviously, handling so many states is nearly impossible, and the vast majority of states will never arise. Therefore, to simplify the model and reduce the number of possible states, we introduce a limitation on the number of keys that are simultaneously pressed to two. This limitation allows us to almost completely describe the process of typing text on the keyboard. The exception is the use of control combinations, which involve the simultaneous pressing of three or more keys. These combinations can be excluded from the model or considered separately as additional features. Thus, taking into account the described limitation, the number of possible states can be calculated as one initial state, when all keys are released, the number of possible states when one key is pressed and the number of states with two simultaneously pressed keys becomes equal to 1712, which is a more acceptable number for constructing models of keystroke dynamics.

Thus, this approach allows us to represent the process of typing on the keyboard not as a sequence of events of pressing or releasing an arbitrary key, but as a sequential change in the states of the keyboard, represented as vectors of a fixed size, each of which is associated with a time stamp. The time intervals between two consecutive states are features on the basis of which a biometric image of the keystroke dynamics can be built. This feature combines the concepts of the interval between keystrokes (FT) and the key holding time (DT).

In this paper, we propose a different approach that would take into account the described models of sequences of keystrokes. For example, consider a keyboard model consisting of only two keys: A and B. Based on this, we present the state matrix for this model, which is shown in Table I. Columns A and B indicate the state of the key: 0 - released, 1 - pressed. The set of states of the individual keys is designated as the S_i state.

Table 1. State Matrix for a Keyboard Model Consising Two Keys

A	B	State
0	0	S_0
0	1	S_1
1	0	S_2
1	1	S_3

In this case, the sequence of states for two consecutive keystrokes without intersection will be as follows: $S_0 \xrightarrow{\Delta t_1} S_2 \xrightarrow{\Delta t_2} S_0 \xrightarrow{\Delta t_3} S_1 \xrightarrow{\Delta t_4} S_0$. Graphically, this process can be represented as shown in Figure 2.

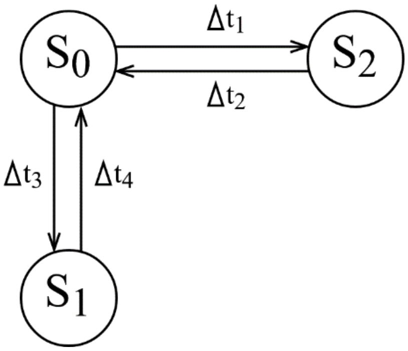

Figure 2. State sequence for a model without intersections

For the second model with incomplete overlap, the sequence will be as follows: $S_0 \xrightarrow{\Delta t_1} S_2 \xrightarrow{\Delta t_2} S_3 \xrightarrow{\Delta t_3} S_1 \xrightarrow{\Delta t_4} S_0$. This process is shown graphically in Figure 3.

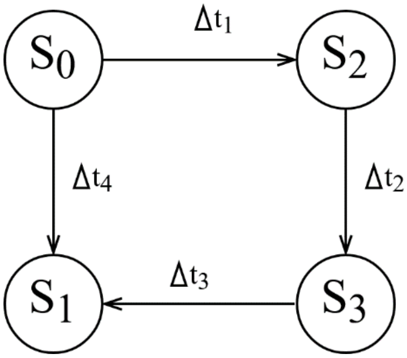

Figure 3. Sequence of states for a partial intersection model

The sequence of states for the full overlap model will look like: $S_0 \xrightarrow{\Delta t_1} S_2 \xrightarrow{\Delta t_2} S_3 \xrightarrow{\Delta t_3} S_2 \xrightarrow{\Delta t_4} S_0$, which is shown in Figure 4.

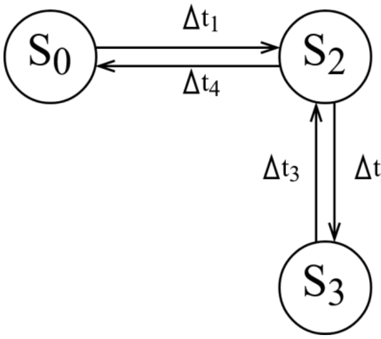

Figure 4. Sequence of states for a model with complete intersection

With a sequential change of states, which corresponds to the process of typing on the keyboard, the final state of each S_0 model in the selected template is the initial state for the next block of states. Therefore, it is proposed to define the context of the chain of states, which contains 4 consecutive states, between which three of time intervals are calculated. Thus, the composite sign of the keystroke dynamics will have the following form: $F_{ijkl} = \left\{\{S_i, S_j, S_k, S_l\}; \{\Delta t_{ji}, \Delta t_{kj}, \Delta t_{lk}\}\right\}$, where i, j, k, l are keyboard state codes.

This approach will allow us not only to extract the primary features of the keystroke dynamics, but also to separate them depending on the used model of the sequence of keystrokes, which were considered earlier. In addition, the presentation of the biometric image of the keystroke dynamics in the form of a state context allows using a single feature space that combines FT, DT and their derivatives. It is assumed that this method will help with building the most complete and accurate model of user keystroke dynamics.

Algorithm for the building of state contexts

As described earlier, the keyboard generates events of the form:

$$Event^{Keycode}{}_i(\text{Timestamp})$$

A sequence of events of this kind is converted into a sequence of state contexts according to the following algorithm.

Let's define a tuple:

$$M = (m_1, m_2, m_3, m_4) \tag{4}$$

Where $m_i = Event^{Keycode}{}_i(\text{Timestamp})$ is the number of consecutive events stored in the tuple corresponds to the number of events in the context.

At the initial moment, this tuple is empty. Each time an event occurs, it is placed in a tuple. Time intervals are calculated between successive events in a tuple and stored in a separate tuple T:

$$T = (\Delta t_{2;1}, \Delta t_{3;2}, \Delta t_{4;3}) \tag{5}$$

After each insertion to the tuple, the content is parsed. If the number of events in a tuple is four, the tuples M and T are transformed into a context of the form $F_{ijkl} = \left\{\{S_i, S_j, S_k, S_l\}; \{\Delta t_{ji}, \Delta t_{kj}, \Delta t_{lk}\}\right\}$ and transmitted for further processing. If the tuple already contains four elements when the m_i event is raised, the elements are shifted to the left, removing the first element. The new event is placed in place of the last item. A similar procedure is performed for a tuple of intervals between events.

$$\begin{gathered} M_{i-1} = (m_{i-4}, m_{i-3}, m_{i-2}, m_{i-1}) \\ M_i = (m_{i-3}, m_{i-2}, m_{i-1}, m_i) \\ T_{i-1} = \left(\Delta t_{i-3;i-4}, \Delta t_{i-2;i-3}, \Delta t_{i-1;i-2}\right) \\ T_i = (\Delta t_{i-2;i-3}, \Delta t_{i-1;i-2}, \Delta t_{i;i-1}) \end{gathered} \tag{6}$$

Thus, the algorithm for building a biometric image based on the presentation in the form of state contexts can be represented in the form of an activity diagram shown in Figure 5.

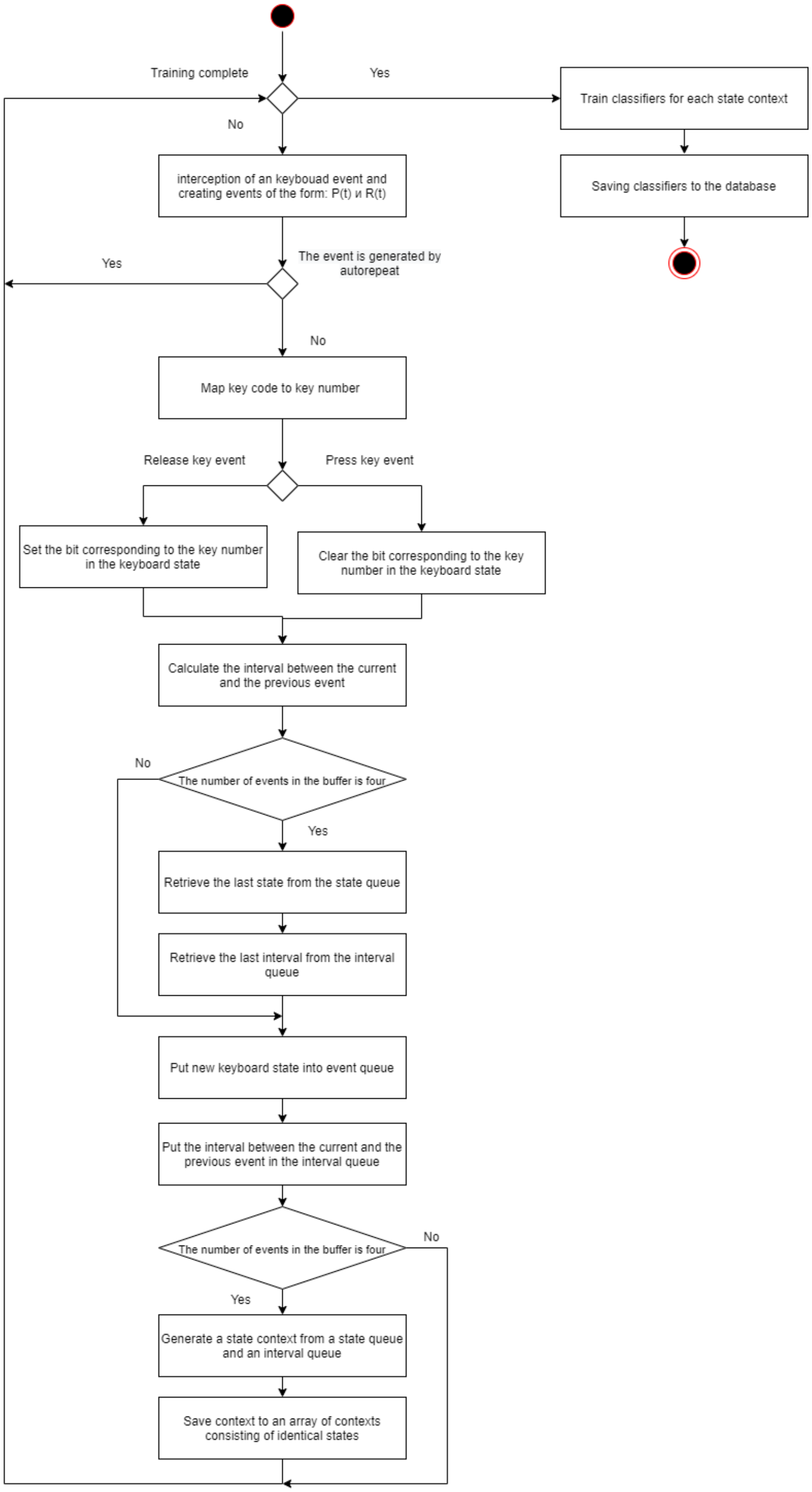
Training complete
Yes
Train classifiers for each state context
No
interception of an keybouad event and creating events of the form: P(t) и R(t)
Saving classifiers to the database
Yes
The event is generated by autorepeat
No
Map key code to key number
Release key event
Press key event
Set the bit corresponding to the key number in the keyboard state
Clear the bit corresponding to the key number in the keyboard state
Calculate the interval between the current and the previous event
No
The number of events in the buffer is four
Yes
Retrieve the last state from the state queue
Retrieve the last interval from the interval queue
Put new keyboard state into event queue
Put the interval between the current and the previous event in the interval queue
No
The number of events in the buffer is four
Yes
Generate a state context from a state queue and an interval queue
Save context to an array of contexts consisting of identical states

Figure 5. Algorithm for the building of a biometric image based on the presentation in the form of state contexts

Benefits of state-based representation of keystroke dynamics

On the basis of the described representation of typing text process, it is possible to describe the feature vector, which will be the basis for the static analysis of the keystroke dynamics. Regardless of which set of features is used to build the feature vector, they can be described as a sequential change of the keyboard state. That is, this approach is a universal way of representing various temporal characteristics of keystroke dynamics: intervals between keystrokes, key holding times, n-graphs or their derivatives.

This format can be used as a basis for static identification. For example, if the vector of intervals between keystrokes is used as the basis of the identification method, then the process of typing a passphrase can be described as a pair of tuples of the following form:

$$\begin{aligned} S &= (S_1, S_2, \dots, S_i \dots, S_N), \nexists S_i = 0 \\ T &= (t_{2\,1}, t_{3\,2} \dots, t_{N\,N-1}), \end{aligned} \tag{7}$$

where N is the number of characters in the passphrase, S is a sequential tuple of states that does not contain zero states, T is a tuple of intervals between the corresponding events.

Thus, the set of methods and approaches of static analysis, which was considered earlier, can be adapted for a universal representation based on the representation based on the state of the keyboard, regardless of what set of features was initially selected for a particular method.

In addition, the described method of representing the keystroke dynamics in the form of a sequential change of keyboard states can be used to perform a dynamic analysis of the keystroke dynamics. For this purpose, the concept of a state context was introduced, which represents four sequential states and the time intervals between them. As noted earlier, four sequential states allow us to fully describe the pressing of two key sequences, taking into account all the described intersection patterns. Within the framework of this study devoted to the development of a for keystroke dynamics analysis system, it is proposed to consider the continuous process of typing text on the keyboard during the entire user session in the form of a sequential change in state contexts. In this case, the state context is used as an independent vector, for each of which an identification procedure is executed using one or more static identification methods. Thus, in this approach, it is proposed to analyze the time window for the sequence as a passphrase, after which the sequence of classification decisions for each individual context is aggregated into a common identification solution.

4. CONCLUSION

Thus, this article has shown that ensuring information security using biometric identification methods, both as an independent means and as a strengthening of existing security systems, is a complex and urgent task.

The existing methods and algorithms for keystroke dynamics analysis were considered, both based on a fixed passphrase and continuous analysis throughout the entire user session. The common ways of representing the temporal characteristics of the analyzed keystroke dynamics were identified.

Within the research on the development of methods for dynamic analysis of keystroke dynamics, a new method of representing the process of typing on the keyboard in the form of a sequential change of states is proposed. On the basis of that representation method the selection of certain sequences, called the context of the state, is proposed. The plurality of state contexts describes the complete biometric image of the user's keystroke dynamics.

One of the advantages of this approach is a generalized representation of all temporal characteristics of the keystroke dynamics, based on the event representation, both basic (FT, DT) and derivatives (n-graphs), which will allow it to be adapted for the vast majority of methods of static user identification by keystroke dynamics.

In addition, the described method of representing keystroke dynamics based on state contexts will significantly expand the set of methods for dynamic user identification based on keystroke dynamics by using a variety of methods for static analysis of keystroke dynamics. This approach, in combination with a generalized representation based on sequential changes in keyboard state, will allow in the future to build productive and flexible keystroke dynamics analysis systems.

CONFLICT OF INTEREST

The authors confirm that the information provided in the article does not contain a conflict of interest.

ACKNOWLEDGMENTS

The reported study was funded by RFBR, project number 19-37-90127.

REFERENCES

Ali, M. L., Thakur, K., Tappert, C. C., & Qiu, M. (2016, June). Keystroke biometric user verification using Hidden Markov Model. In *2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud)* (pp. 204-209). IEEE.

Bhatia, A., & Hanmandlu, M. (2017). Keystroke dynamics based authentication using information sets. *Journal of Modern Physics*, *8*(09), 1557.

Bours, P., & Mondal, S. (2015). Continuous Authentication with Keystroke Dynamics. 10.13140/2.1.2642.5125.

Bryukhomitsky, U.A. (2010). keystroke recognition based on using statistical density estimates distribution. Information countermeasures against terrorism threats. *Scientific and practical journal, 14*, 149 – 154.

Daniele Gunetti and Claudia Picardi. (2005). Keystroke analysis of free text. ACM Trans. Inf. Syst. Secur., 8(3), 312–347.

Dowland, P. (2001). A preliminary investigation of user authentication using continuous keystroke analysis. In *Proc IFIP Annual Working Conf on Information Security Management and Small System Security, 2001* (pp. 27-28).

Dowland, P. (2001). A preliminary investigation of user authentication using continuous keystroke analysis. In *Proc IFIP Annual Working Conf on Information Security Management and Small System Security, 2001* (pp. 27-28).

Dowland, P. S., & Furnell, S. M. (2004, August). A long-term trial of keystroke profiling using digraph, trigraph and keyword latencies. In *IFIP International Information Security Conference* (pp. 275-289). Springer, Boston, MA.

Dowland, P. S., Furnell, S. M., & Papadaki, M. (2002). Keystroke analysis as a method of advanced user authentication and response. In *Security in the Information Society* (pp. 215-226). Springer, Boston, MA.

Dowland, P.S., & Furnell, S.M. (2004). A long-term trial of keystroke profiling using digraph, trigraph and keyword latencies. In Security and Protection in Information Processing Systems, volume 147 of IFIP The International Federation for Information Processing, pages 275–289. Springer US.

Gaines, R.S., Lisowski, W., Press, S.J., & Shapiro, N. R-2526-NSF. Santa Monica, Calif, USA: Rand Corporation; 1980. Authentication by keystroke timing: some preliminary results.

Gunetti, D., & Picardi, C. (2005). Keystroke analysis of free text. *ACM Transactions on Information and System Security, 8*(3), 312–347.

Joyce, R., & Gupta, G. (1990). Identity authentication based on keystroke latencies. *Communications of the ACM*, *33*(2), 168-176.

Killourhy, K. S., & Maxion, R. A. (2009, June). Comparing anomaly-detection algorithms for keystroke dynamics. In *2009 IEEE/IFIP International Conference on Dependable Systems & Networks* (pp. 125-134). IEEE.

Killourhy, K., & Maxion, R. (2008, September). The effect of clock resolution on keystroke dynamics. In *International Workshop on Recent Advances in Intrusion Detection* (pp. 331-350). Springer, Berlin, Heidelberg.

Kobojek, P., & Saeed, K. (2016). Application of recurrent neural networks for user verification based on keystroke dynamics. *Journal of telecommunications and information technology*, (3), 80-90.
Loy, C. C., Lai, W. K., & Lim, C. P. (2007, November). Keystroke patterns classification using the ARTMAP-FD neural network. In *Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007)* (Vol. 1, pp. 61-64). IEEE.

Mandujano, S., & Soto, R. (2004, September). Deterring password sharing: User authentication via fuzzy c-means clustering applied to keystroke biometric data. In *Proceedings of the Fifth Mexican International Conference in Computer Science, 2004. ENC 2004.* (pp. 181-187). IEEE.

Monrose, F., & Rubin, A. (1997, April). Authentication via keystroke dynamics. In *Proceedings of the 4th ACM Conference on Computer and Communications Security* (pp. 48-56).

Pashchenko, D.V., Balzannikova, E.A., Sergina, I.G. (2018). User identification method by means of biometric image of keystroke dynamics with double-chained representation. *Вопросы радиоэлектроники, 12*, С 83–89. DOI 10.21778/2218-5453-2018-12-83-89.

Pavaday, N., & Soyjaudah, K. M. S. (2007, September). Investigating performance of neural networks in authentication using keystroke dynamics. In *AFRICON 2007* (pp. 1-8). IEEE.

Pavaday, N., & Soyjaudah, K. M. S. (2008, October). A comparative study of secret code variants in terms of keystroke dynamics. In *2008 Third International Conference on Risks and Security of Internet and Systems* (pp. 133-140). IEEE.

Plank, B. (2016). Keystroke dynamics as signal for shallow syntactic parsing. *arXiv preprint arXiv:1610.03321*.

Sheng, Y., Phoha, V. V., & Rovnyak, S. M. (2005). A parallel decision tree-based method for user authentication based on keystroke patterns. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *35*(4), 826-833.

Shepherd, S. J. (1995). Continuous authentication by analysis of keyboard typing characteristics. In European Convention on Security and Detection, pages 111–114.

Sidorkina, I.G., & Savinov, A.N. (2013). Three algorithms of control access to the KSSI on the basics of recongnition of keystroke dynamics. *Bulletin of the Chuvash University, 3,* 293-301.

Song, D., Venable, P., & Perrig, A. (1997). User recognition by keystroke latency pattern analysis. *Retrieved on*, *19*.