Trace-Based Authentication Biometric for On-Line Education

Fatma Derbel Bouattour and Pierre-Antoine Champin
Université de Lyon, CNRS Université Lyon 1, LIRIS, UMR5205, F-69622, France

Keywords: Authentication, Biometric, Keystroke Dynamics, Mouse Dynamics, m-Traces, e-Learning

Abstract: In this paper, we present a continuous behavioral authentication system for Web applications and in particular for on-line education applications. Our system implement two behavioral biometric modalities, keystroke dynamics and mouse dynamics, with m-traces concepts. We describe our trace based authentication approach as well as the experiments we have conducted.

1 INTRODUCTION

On-line educational engineering is facing numerous challenges and issues linked to teaching courses and knowledge transmission, this evolution is encountering much unwillingness when granting and recognizing diplomas or qualifications. In fact, many organizations don't give equivalent credits for qualifications obtained on-line compared to those obtained in situ in a school or a learning center. One of the main arguments for this lack of trust in remote learning programs is the fact that users may share authentication credentials in order to cheat and obtain the certification. So for organizations that use these platforms to award qualifications, the question of the authentication's reliability is crucial because it guarantees the credibility of the delivered certificates.

This paper focuses on the issue of learners authentication in Web-based e-learning platforms. Authentication is the process of confirming a user's identity, usually before giving them access to resources or services in a secure environment. Many user authentication approaches have been proposed in the literature, and they can be divided into three categories (O'Gorman, 2003): knowledge-based authentication, object-based authentication and biometric authentication.The biometrics authentication methods are divided into two categories: physiological biometrics (face, fingerprint,..) and behavioral biometrics (keystroke dynamics, mouse dynamics,..) (Matyas and Riha, 2003).

Biometric authentication has several advantages over knowledge-based and object-based authentication because biometrics cannot be forgotten, stolen, or misplaced. Additionally, behavioral biometric authentication methods are less obtrusive than other biometric methods and do not require special hardware in order to capture the necessary biometric data.

Authentication methods are also divided into static and continuous methods. A static authentication system authenticates the user only once, when they open their session, so it can not detect any change of identity during a session. On the other hand, a continuous authentication system verifies the user's authenticity during the whole session. There are many ways to implement continuous authentication systems, but behavioral biometrics seems a good choice due to the unobtrusive data collection that it allows. While keystroke dynamics (KD) and mouse dynamics (MD) are commonly used for this purpose (see Section 2 for more details on these two modalities), we argue that complementing them with other behavioral indicators can help improve the robustness of continuous authentication systems. Of course, this requires a data model that is versatile enough to capture multiple facets of the user's activity, such as the m-trace metamodel proposed by (Mille et al., 2013) (described in more details in Section 3.1).

In this paper, we present a continuous authentication system based on behavioral data implemented with m-trace, which we have designed for for the context of e-learning Web applications. This paper describes the following contributions:

- We propose a general architecture of an authentication system based on interaction traces.

- We describe how biometric behavior modalities can be implemented in our authentication system and integrated it in an e-learning platform in order to continuously and dynamically authenticate learners.

- We compare the proposed system with state-of-

Bouattour, F. and Champin, P.
Trace-Based Authentication Biometric for On-Line Education.
DOI: 10.5220/0011776200003470
In *Proceedings of the 15th International Conference on Computer Supported Education (CSEDU 2023) - Volume 1*, pages 154-161
ISBN: 978-989-758-641-5; ISSN: 2184-5026

the-art authentication approaches in the context of e-learning platforms.

The remainder of this paper will be as follows. In the next section, we present a state of the art of authentication systems implementing KD and MD in e-learning platform. In the Section 3, we describe the methodology followed in our research. In Section 4, we describe the operation of each component of our authentication system for both the KD and MD modalities. Experimental results are described and analyzed in Section 5. Finally, we conclude and present some further developments of this research.

2 RELATED WORKS

Continuous Behavioral Biometric Authentication Systems Using KD and MD. In this section, we present the state of the art in continuous behavioral biometric authentication systems , more specifically on those based on the KD and MD modalities.

The motivation behind the use of these behavioral biometric data is that they do not require any additional hardware for data capture. In addition, they can be collected without interrupting the user's activity.

Most researches on this topic report performances in terms of False Acceptance Rate (FAR), False Recognition Rate (FRR), and sometimes Equal Error Rate (EER) (Ahmed and Traore, 2005; Pusara, 2007; Bailey et al., 2014). FAR and FRR are defined as percentages; FAR represents the chances that an impostor is accepted by the system, while FRR represents the chances that a genuine user is rejected by the system. The EER is the value where FAR and FRR are equal. Other researches (Mondal and Bours, 2017) have used different indicators for measuring authentication performance: they focus on the number of actions that impostors and genuine users, respectively, can perform before being rejected or terminating the session. The Average Number of Impostor Actions (ANIA) should therefore be as low as possible, while the Average Number of Genuine Actions (ANGA) should be as high as possible.

Researches on continuous authentication using KD was started by (Shepherd, 1995). They authenticate users by analyzing the way they type on a keyboard (Bergadano et al., 2002). Some approaches require the user to type a fixed text (Kolakowska, 2011) while others can work on free text (Kang and Cho, 2015). Most existing researches on KD use the time between two successive keys as features (Araújo et al., 2004), although some researches also consider n-graph durations (Davoudi and Kabir, 2010). For classifying users, many researches compare feature vectors using various distance measures (Bours, 2012), or other measures of how their order differ (Kolakowska, 2011). Others use machine learning techniques, such as nearest neighbor classifiers (Kang and Cho, 2015) and neural networks (Ahmed and Traore, 2014). Another approach for continuous authentication, based on a trust model, has been proposed by (Bours and Mondal, 2015). The confidence in the user's authenticity is represented by a trust value, which is revised for each action by comparing it to the user profile, and applying a penalty-and-reward function.

Despite a few works using MD for continuous authentication (Shen et al., 2013), this kind of biometric data has not been used as much as KD. (Gamboa and Fred, 2004) propose an authentication system that collects mouse move actions from a Web interface. Ahmed and Traore (Ahmed and Traore, 2007) use artificial neural networks to model the users' behavioral characteristics from the captured data.

Mondal and Bours (Mondal and Bours, 2013) use the trust model presented above to propose a continuous authentication system based on MD behavioral data.

Several researches have used a combination of KD and MD in a continuous authentication system (Ahmed and Traore, 2005; Pusara, 2007; Bailey et al., 2014). Indeed, since only one biometric modality is often insufficient to robustly verify the user's identity, a combination of multiple modalities can increase the performance of the authentication system (Ross and Jain, 2003).

Biometric Authentication System in e-Learning. In this section, we present an overview of most relevant published works on the usage of biometric authentication in eLearning systems. Many of this authentication methods can be applied in e-Learning systems in order to prevent impostors from accessing teaching resources (Belashenkova et al., 2015).

(Agulla et al., 2008) presents a biometric authentication system based on face recognition in the form of a web application. This application can be easily integrated into LMS, and allows the use of facial recognition during access control, tracking and assessment. (Fayyoumi and Zarrad, 2014) developed also an authentication system for on-line exams using face recognition. Facial authentication is generally a reliable method for identification, but the variation of the qualities of web-cams in addition to the lighting conditions in the learners' environment do not always allow an image of sufficient quality to authenticate them.

(Rabuzin et al., 2006) recommended the system of

multi-biometric methods. The authors argued that authentication based on a single biometric method could cause security risks. Another multi-biometric authentication system for eLearning systems was introduced by (Traoré et al., 2017). (Asha and Chellappan, 2008) suggested to combine MD and fingerprint recognition using a mouse with an inbuilt fingerprint sensor. (Traoré et al., 2017) proposed a framework combines keystroke mouse and face biometric for continuous authentication.

Further approaches for continuous multi-biometric authentication in eLearning systems have been presented in cite (Hernandez-Ortega et al., 2019). Others researchers recommended a combination of biometric technologies analysing face, voice, mouse and keystroke dynamics (Jagadamba et al., 2020).

(Flior and Kowalski, 2010) suggested to employ continuous authentication based on KD for written exams. (Morales and Fierrez, 2015) employ KD to authenticate learner during a real on-line exam, testing their system with 64 subjects. (Ramu and Arivoli, 2013) propose an authentication system to improve on-line examination by utilizing KD and knowledge based authentication. In this system exam participant are authenticated only statically at login time. (Pleva et al., 2016) demonstrate that using keystroke and mouse dynamics is one of the best choices for continuously authentication in current e learning systems.

(Maas et al., 2014) present the Signature Track authentication process used in the e-learning platform Coursera. The process uses two biometric authentication approaches, face recognition and KD. the Coursera platform asks the learner to provide a photo with web-cam of his or her face as well as a photo of a government-issued ID document, and to establish a keystroke biometric profile by typing a short phrase. But this method of authentication is not continuous since it authenticates the learners at the end of a course or exercise. In addition it is invasive since it interrupts the learning activity to perform the authentication process.

3 GENERIC MULTI-MODAL AUTHENTICATION

3.1 The m-Trace Meta-Model

In order to take into account multiple behavior modalities, we need a flexible model to capture the interactions of the user with the system. We propose to use the meta-model defined by (Mille et al., 2013), as it provides the desired flexibility.

The central notion of the meta model is that of **m-trace** (short for "modeled trace"), which is a sequence of records representing the actions performed by a user of a system. Each such record is called an **obsel** (short for "observed element"), and is described by a type, a set of attributes, and two timestamps (*begin* and *end*, delimiting the time interval in which the obsel occurred[1]). Each m-trace is associated with a **trace model** which specifies the obsel types that trace may contain, as well as the attributes obsels of each type can have. The trace model allows to elicit the structure and the underlying semantics of the m-trace. M-traces are stored and processed in a system called a Trace-Base Management System (TBMS). TBMSs contain two kinds of m-traces: **primary traces** contain the obsels collected directly from the applications; **transformed traces**, on the other hand, are computed by processing the obsels of one or several source traces (which can be either primary or transformed). Transformations are typically used to "lift" the description of the activity from a low-level trace model (focusing on atomic interactions) to a higher-level one (describing more abstract actions and tasks).

The system presented in this paper uses kTBS[2], a reference TBMS implementation providing a number of useful transformation operators, from simple filters to more complex rewriting methods, based on the SPARQL query language (Harris and Seaborne, 2013) or finite state automata (FSA).

3.2 System Architecture

We describe in this section the architecture of our biometric authentication system based on m-traces.

Users' interactions with the platform are collected by the collection system **Tracing_You**[3], a simple script that can be added to any Web application, and configured to collect a wide range of events inside that application. Those events are then sent as obsels to kTBS, which stores them in a primary trace (Cordier et al., 2015).

These events must then be processed by the **Data Analysis Module** to calculate the biometric features and measures for each modality. The data analysis module converts the collected atomic events into higher-level obsels describing user *actions*. This module is divided in sub-modules, each one handling a specific modality. For the moment, a sub-module for KD and another one for MD are implemented (which will be described in detail in Section 4). They use a set

[1]It is still possible for an obsel to have the same *begin* and *end*, when it represents an instantaneous event.

[2]http://tbs-platform.org/ktbs/

[3]http://tbs-platform.org/tbs/doku.php/tools:tracingyou

of FSA, SPARQL and fusion transformations. FSA transformations are used to detect sequences of obsels matching a given pattern, and produce a new obsel for each occurrence of that pattern. SPARQL transformation use the SPARQL query language (Harris and Seaborne, 2013) to detect more complex arrangements of obsels, and compute new attributes. Fusion transformations merge the content of several sources into a single m-trace. The features computed by the Data Analysis Module are stored in transformed traces, which will then be used by the **Behavior Tracing Module** to build the profile trace, and by the **Signature Creation Module** to build the signature trace.

The profile trace is a transformed trace that describe the learning behavior resulting from different authentication methods (in this article biometric behavior). **The signature trace** is also a transformed trace that contains unique features and statistics extracted from actions after enough occurrences of each type of action have been observed (the threshold is currently set to 5). Finally, the **Identity Decision Module** will compare an profile trace with the signature trace with the signature trace generated from previous (trusted) actions, in order to authenticate the user.

4 IMPLEMENTING KD AND MD

4.1 Data Collection Module

For KD, we use to capture each key press and key release events captured by the Tracing_You module. We therefore define the following obsel types for the primary trace:

- *K_Press* (key press): this type of obsel describes the press of a given key from the keyboard. It contains the following attributes: the character corresponding to the key pressed, if any (*character*), the numeric code for that key (*keycode*) and the time of the event in milliseconds (*begin* and *end* are equal, as obsels of this type are always instantaneous).

-*K_Release* (key release): this type of obsel describes the release of a given key from the keyboard. It contains the same attributes as type K_Press.

For MD, there are three types of events we need to collect, captured by the following obsel types:

-*M_Move* (mouse move): this obsel type describes a point traversed by the mouse pointer while it is moving. It contains the following attributes: the screen coordinates of the cursor (*posX, posY*), and the time of the event (*begin* and *end* are equal).

-*M_BPress* (mouse button press): this obsel type describes the press of a mouse button. It contains the following attributes: the name of the button being pressed (*TypeButton*), the position of the mouse pointer (*posX, posY*), and the time of the event (*begin* and *end* are equal).

-*M_BRelease* (mouse button release): this obsel type is similar to the previous one, but it describes the release of a mouse button.

4.2 Data Analysis Module

The collected data must then be processed by the Data Analysis Module to calculate the biometric features for each modality KD and MD. This module is divided in sub-modules, a sub-module for KD and another one for MD. They use a set of finite state automate (FSA), SPARQL and fusion transformations. FSA transformations are used to detect sequences of obsels matching a given pattern, and produce a new obsel for each occurrence of that pattern.

For KD, various features can be extracted from the raw data. We use in this research the most commonly used in the literature (Zhong et al., 2012): the hold time of a key (PR), the time between the release of a key and the press of the next one (RP), the elapsed time between two consecutive key releases (RR), and the elapsed time between two consecutive key presses (PP). We apply an FSA transformation to calculate these features from the obsels of the primary trace described above. The result of this transformation is a transformed trace that contains the following obsel types:

-*K-PR* (key press-release): this obsel type has the same attributes as *K_Press* above, but its *begin* timestamp corresponds to the key press, while its *end* timestamp corresponds to its release.

- *K-PP* (keys press-press): this obsel type contains the following attributes: the characters corresponding to the first key pressed (*charSource*) and the next one (*charDestination*), the numeric key code of the first key pressed (*keySource*) and the second one (*keyDestination*); its *begin* timestamp corresponds to the first key press, while its *end* timestamp corresponds to the second one.

- *K-RR* (keys release-release) and K-RP (keys release-press): these types of obsel are similar to the previous one, but capture the release time of first key in *begin* and the release (resp. press) of the second one in *end*.

For MD, we use four types of actions defined in the literature (Mondal and Bours, 2013): move, drag-and-drop, single click and double click.

As with KD, we apply an FSA transformation to calculate these features from the three kinds of obsels in the primary trace. The result of this transformation

is a Transformed trace that contains the following obsel types:

- *M-BPR* (mouse button press-release): this obsel type describes a single click. Its obsels are produced whenever a mouse button is pressed and released in less than a given time (currently set to 2000ms). They have the following attributes: the name of the button being clicked (*TypeButton*), the position of the click in the screen (*xSource*, *ySource*), its *begin* timestamp corresponds to the button press, while its *end* timestamp corresponds to its release.

- *M-BDC* (mouse button double click): this obsel type describes a double click. Its obsels are produced whenever two clicks are separated by less than a given time (currently set to 1000ms). The have the same attributes as *M-BPR* above, but their *begin* timestamp corresponds to the first press, while its *end* timestamp corresponds to the second release.

- *M-MS* (mouse move sequence): this obsel type describes continuous movements of the mouse pointer. Its obsels are produced by sequences of *M-Move* obsels with less than 250ms between them. A combination of four transformations (FSA and SPARQL) is used to compute the following attributes: the time when the movement began (*begin*) and ended (*end*), the position where the cursor started (*xSource, ySource*) and finished (*xDestination, yDestination*), the straight-line distance between those two points (*traveledDistance*), the length of the pointer movement (*curveLength*), the average speed (*curveSpeed*), and the overall direction of the movement between 8 possible directions (*direction*).

- *M-DD* (mouse drag and drop): this obsel type represents a movement of the mouse pointer started when pressing a button, and ended when the button is released. It has the same attributes as both *M-MS* and *M-BPR*.

4.3 Behavior Tracing Module

The result of the behavior tracing module is a transformed trace called the *profile trace* that contains action information for each of the KD and MD modalities. This trace allows to describe the behavior of the user: their way of typing on the keyboard and their way moving the mouse pointer. The profile trace then contains these types of obsels (*K-PR, K-PP, K-RR, K-RP, M-MS, M-DD, M-BPR, M-BDD*).

4.4 Signature Creation Module

A signature in our authentication system is a transformed trace that contains unique features extracted from each user action after it enough times (the repetition number is set to 5 in this study). Our authentication system builds a signature trace for each authentication modality.

The signature trace for KD is produced by a SPARQL transformation computing aggregated information from the profile trace, resulting in the following obsel types:

- *K-SPR*: an obsel of this type is created for each key, aggregating a set of trusted *K-PR* obsels on that key. It has the following attributes: the number of repetitions of the action, and the mean and standard deviation of their durations ($end - begin$).

- *K-SRR*: an obsel of this type is created for each pair of successive keys, aggregating a set of trusted *K-RR* obsels on that key pair. It has the following attributes: the number of repetitions of the action, and the mean and standard deviation of their durations.

- *K-SPP*, *K-SRP*: these obsel types are similar to *K-SRR* above, but aggregating *K-PP* and *K-RP*, respectively.

As for KD, the signature trace for MD is produced by a SPARQL transformation computing aggregated information from the profile trace, resulting in the following obsel types:

- *M-SBPR* and *M-SBDC*: an obsel of each of these types is created, aggregating a set of trusted *M-BPR* and *M-BDC* obsels, respectively. It has the following attributes: the number of repetitions of the action, and the mean and standard deviation of their durations.

- *M-SMS* and *M-SDD*: an obsel of each of these types is created for each direction, grouping a set of trusted *M-MS* and *M-DD* obsels, respectively. It has the following attributes: the number of repetitions of the action, and the mean and standard deviation of their durations, traveled distance, curved length, speed and acceleration.

4.5 Identity Decision Module

As in other biometric systems, we need to determine the distance between the signature trace and the new input (each action in the profile trace in our system). We use in our system a scaled Manhattan distance. For KD obsels (and click-related MD obsels), we compare each action obsel to the signature obsel of the corresponding type, and related to the same pair of keys (or same button). The action obsel has a *duration* ($end - begin$), and the signature obsel contains the mean m and standard deviation sd of the time in genuine actions. The distance of the action to the user's signature is therefore defined as:

$$D = \frac{|m - duration|}{sd} \quad (1)$$

For move-related MD obsels, action obsels contain a number of attributes $value_i$ (acceleration, speed, traveled distance), and signature obsels contain the mean m_i and standard deviation sd_i for each of those attributes. The distance between them is therefore defined as:

$$D = \frac{1}{n}\sum_{i=1}^{n}\frac{|m_i - value_i|}{sd_i} \quad (2)$$

Once the system has determined the distance D between the current action and the signature, then this value must be used to update a trust value, in a way that is freely inspired by the the trust model used in (Mondal and Bours, 2017). The penalty and reward function used to calculate the trust is the following:

$$Trust = \begin{cases} 0, & \text{initially} \\ Trust + D_{threshold} - D & \text{if } D \leq D_{noise} \\ Trust & \text{if } D > D_{noise} \end{cases} \quad (3)$$

The confidence value starts at zero (no opinion about the authenticity of the user). Then it will increase (positive opinion) or decrease (negative opinion) based on the distance of each action. More precisely, a small distance (smaller than a threshold $D_{threshold}$) will reward the user by increasing the trust, while a higher distance will penalize them, decreasing the trust. Very high distances (higher than D_{noise}) are considered as irrelevant outliers, and therefore do not impact the trust.

5 EXPERIMENTAL RESULT

In this section we describe the results of an experiment we have conducted to assess the performances of our system. The experiment was conducted with 23 office colleagues. The participants were asked to play a Web-based puzzle game, requiring them to drag and drop images across a page, and to type a given text. This application can collect a primary m-trace complying with the trace models described in Sections 4.1. The experiment was divided into two phases. In the enrollment phase, the collected data was used to build the user's signature trace. In the authentication phase, the users were asked to replay the same game and the collected data was used to calculate the trust value for each action. Then, to simulate impostor behaviors, we re-used each user's second trace with each other user's signature. This provided us with 23 cases of genuine user, and $23 \times 22 = 506$ cases of impostor.

The evolution of trust for typical genuine user is a quick increase of value during the first action, after which the trust stays well above zero. On the other hand, for most simulated impostors, the trust continuously decreases, therefore remaining below zero.

Table 1: Influence of $D_{threshold}$ on FAR and FRR pour.

$D_{threshold}$	1	1.2	1.5
FAR	0%	0%	4.34%
FRR	100%	49.4%	0%

We consider that a user is genuine if their final trust (after all actions) is positive, and that they are an impostor if their final trust is negative.

To evaluate our algorithm, we start by empirically determining D_{noise} and $D_{threshold}$.

To determine the value of D_{noise}, we studied the distribution for each user of the distances of all actions to their respective signature. We noticed a significant similarity in the distributions, with values varying between 0 and 54. We noticed then the value 17 represents the 90th percentile of our whole dataset. We therefore decided to set $D_{noise} = 17$.

To determine the value of $D_{threshold}$, we computed the trust of all genuine users and impostors with three different $D_{threshold}$ (1, 1.2, 1.5), in order to compute the FAR and FRR. The results are given in Table 1. Our priority being to avoid rejecting genuine users (low FRR), we focus on $D_{threshold} = 1.5$. However, we also used Table 1 to compute the EER of our method, as some related works only provide the EER. The ERR value of our authentication method is 4.016%.

With $D_{threshold} = 1.5$, our system correctly recognized all 23 genuine users, giving a False Reject Rate (FRR) of 0%. It wrongly considered as genuine 22 impostors out of 506, giving a False Acceptance Rate (FAR) of 4.34% and a Equal Error Rate (ERR) of 4.016%. These results are among the best results comparing with other works combining KD and MD, as can be seen in the table 2. We did not compute the ANIA and ANGA indicators (described in Section 2) because they do not apply to our context where presumed impostors are never blocked (as explained in the previous section).

6 CONCLUSION AND PERSPECTIVES

In this paper, we have presented a continuous behavioral authentication system based on KD and MD. An experiment conducted on 23 users showed that the performances of our system are comparable to similar works from the literature. We have answered the research questions stated in our introduction by demonstrating that: the keyboard and mouse events collected by a Web browser are sufficient to authen-

Table 2: Performances of authentication system mixing KD and MD (N is the number of participants).

	FAR (%)	FRR (%)	ERR (%)	N
(Ahmed and Traore, 2005)	0.65	1.31		22
(Pusara, 2007)	14.47	1.78		61
(Bailey et al., 2014)	2.24	2.10		31
(Pleva et al., 2016)			4.5	50
(Morales and Fierrez, 2015)			6.07	64
(Jagadamba et al., 2020)				
Our approach	4.34	0.00	4.16	23

ticate users with reasonable performances (compared to similar related works); and that the m-trace meta-model (Mille et al., 2013) can be used to implement a continuous behavioral authentication system.

This opens a number of interesting perspectives for future works. First, the architecture of our system is naturally extensible to other behavioral data, thanks to the genericity of the m-trace meta-model. In the domain of e-learning, specifications such as SCORM (Bohl et al., 2002) and xAPI (ADL, 2013) provide standard ways for learning activities to expose traces of the user's activity. We are currently considering using these traces as a source of high-level behavioral data which could complement the lower-level KD and MD data in the authentication process. This is all the more important that KD and MD data are probably very sensitive[4] to hardware changes (such as people switching from an external mouse to a touchpad), while higher-level behaviors may be more robust.

Second, even without considering hardware changes, we expect that the KD and MD of a given user will evolve over time (as the user gets used to a given keyboard, or simply improves their typing skills, for example). In order to improve our system into a truly *dynamic* authentication system, we need to update the user's signature with the most recent trusted profile traces. In the future, we will study the design of such update mechanisms. The challenge is, of course, to prevent an impostor's profile trace to pollute the updated signature.

Finally, and related to the previous point, we would like to refine the decision criteria used to discriminate genuine users from impostors. Although the current approach (using the sign of the final trust) gives good results, observing in more detail how the trust evolves over time raises interesting questions. For example, Figure 1 represents the evolution of trust

[4]To the best of our knowledge, there has been no work in the literature studying this sensitivity. Some data in our own experiment seem nonetheless to confirm this intuition.

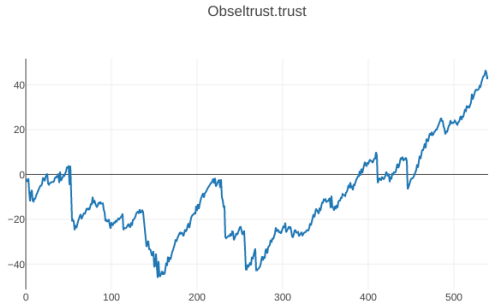

Figure 1: Evolution of trust for a genuine user against the number of actions.

for a genuine user, which was eventually correctly recognized as such. We can see that, should the session have stopped earlier, this user may have been wrongly considered an impostor. We are planning to study other indicators, taking into account the whole sequence of trust values rather than just the last one.

REFERENCES

ADL (2013). Experience API v1.0.3. Technical report, Advance Distributed Learning (ADL) Initiative, Department of Defense. https://github.com/adlnet/xAPI-Spec.

Agulla, E. G., Rifón, L. A., Castro, J. L. A., and Mateo, C. G. (2008). Is my student at the other side? applying biometric web authentication to e-learning environments. In *Advanced Learning Technologies, 2008. ICALT'08. Eighth IEEE International Conference on*, pages 551–553. IEEE.

Ahmed, A. A. and Traore, I. (2014). Biometric recognition based on free-text keystroke dynamics. *IEEE transactions on cybernetics*, 44(4):458–472.

Ahmed, A. A. E. and Traore, I. (2005). Anomaly intrusion detection based on biometrics. In *Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC*, pages 452–453. IEEE.

Ahmed, A. A. E. and Traore, I. (2007). A new biometric technology based on mouse dynamics. *IEEE Transactions on dependable and secure computing*, 4(3).

Araújo, L., HR Sucupira, L., Lizárraga, M., Ling, L., and Yabu-uti, J. (2004). User authentication through typing biometrics features. *Biometric Authentication*, pages 101–103.

Asha, S. and Chellappan, C. (2008). Authentication of e-learners using multimodal biometric technology. In *2008 International Symposium on Biometrics and Security Technologies*, pages 1–6. IEEE.

Bailey, K. O., Okolica, J. S., and Peterson, G. L. (2014). User identification and authentication using multimodal behavioral biometrics. *Computers & Security*, 43:77–89.

Belashenkova, N. N., Cherepovskaya, E. N., Lyamin, A. V., and Skshidlevsky, A. A. (2015). Protection methods of assessment procedures used in e-learning. In

Emerging eLearning Technologies and Applications (ICETA), 2015 13th International Conference on, pages 1–6. IEEE.

Bergadano, F., Gunetti, D., and Picardi, C. (2002). User authentication through keystroke dynamics. *ACM Transactions on Information and System Security (TISSEC)*, 5(4):367–397.

Bohl, O., Scheuhase, J., Sengler, R., and Winand, U. (2002). The sharable content object reference model (scorm)-a critical review. In *Computers in education, 2002. proceedings. international conference on*, pages 950–951. IEEE.

Bours, P. (2012). Continuous keystroke dynamics: A different perspective towards biometric evaluation. *Information Security Technical Report*, 17(1):36–43.

Bours, P. and Mondal, S. (2015). Continuous authentication with keystroke dynamics. *Norwegian Information Security Laboratory NISlab*, pages 41–58.

Cordier, A., Derbel, F., and Mille, A. (2015). Observing a web based learning activity: a knowledge oriented approach. Research Report tweak_am_fd_ac_01, LIRIS UMR CNRS 5205.

Davoudi, H. and Kabir, E. (2010). Modification of the relative distance for free text keystroke authentication. In *Telecommunications (IST), 2010 5th International Symposium on*, pages 547–551. IEEE.

Fayyoumi, A. and Zarrad, A. (2014). Novel solution based on face recognition to address identity theft and cheating in online examination systems. *Advances in Internet of Things*, 4(02):5.

Flior, E. and Kowalski, K. (2010). Continuous biometric user authentication in online examinations. In *2010 seventh International Conference on information technology: new generations*, pages 488–492. IEEE.

Gamboa, H. and Fred, A. (2004). A behavioral biometric system based on human-computer interaction. In *Proceedings of SPIE*, volume 5404, pages 381–392.

Harris, S. and Seaborne, A. (2013). SPARQL 1.1 Query Language. W3c Recommendation, W3C.

Hernandez-Ortega, J., Daza, R., Morales, A., Fierrez, J., and Ortega-Garcia, J. (2019). edbb: Biometrics and behavior for assessing remote education. *arXiv preprint arXiv:1912.04786*.

Jagadamba, G., Sheeba, R., Brinda, K., Rohini, K., and Pratik, S. (2020). Adaptive e-learning authentication and monitoring. In *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, pages 277–283. IEEE.

Kang, P. and Cho, S. (2015). Keystroke dynamics-based user authentication using long and free text strings from various input devices. *Information Sciences*, 308(Supplement C):72 – 93.

Kolakowska, A. (2011). User authentication based on keystroke dynamics analysis. *Computer Recognition Systems 4*, pages 667–675.

Maas, A., Heather, C., Do, C. T., Brandman, R., Koller, D., and Ng, A. (2014). Offering verified credentials in massive open online courses. *Ubiquity*, 2014(May):2.

Matyas, V. and Riha, Z. (2003). Toward reliable user authentication through biometrics. *IEEE Security & Privacy*, 99(3):45–49.

Mille, A., Champin, P.-A., Cordier, A., Georgeon, O., and Lefevre, M. (2013). Trace-Based Reasoning - Modeling interaction traces for reasoning on experiences. In McCarthy, P., editor, *The 26th International FLAIRS Conference*, pages 1–15, United States.

Mondal, S. and Bours, P. (2013). Continuous authentication using mouse dynamics. In *Biometrics Special Interest Group (BIOSIG), 2013 International Conference of the*, pages 1–12. IEEE.

Mondal, S. and Bours, P. (2017). A study on continuous authentication using a combination of keystroke and mouse biometrics. *Neurocomputing*, 230:1–22.

Morales, A. and Fierrez, J. (2015). Keystroke biometrics for student authentication: A case study. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE '15, pages 337–337, New York, NY, USA. ACM.

O'Gorman, L. (2003). Comparing passwords, tokens, and biometrics for user authentication. *Proceedings of the IEEE*, 91(12):2021–2040.

Pleva, M., Bours, P., Hladek, D., and Juhar, J. (2016). Using current biometrics technologies for authentication in e-learning assessment. In *Emerging eLearning Technologies and Applications (ICETA), 2016 International Conference on*, pages 269–274. IEEE.

Pusara, M. (2007). *An examination of user behavior for user re-authentication*. PhD thesis, Purdue University.

Rabuzin, K., Baca, M., and Sajko, M. (2006). E-learning: Biometrics as a security factor. In *2006 International Multi-Conference on Computing in the Global Information Technology-(ICCGI'06)*, pages 64–64. IEEE.

Ramu, T. and Arivoli, T. (2013). A framework of secure biometric based online exam authentication: An alternative to traditional exam. *Int J Sci Eng Res*, 4(11):52–60.

Ross, A. and Jain, A. (2003). Information fusion in biometrics. *Pattern recognition letters*, 24(13):2115–2125.

Shen, C., Cai, Z., Guan, X., Du, Y., and Maxion, R. A. (2013). User authentication through mouse dynamics. *IEEE Transactions on Information Forensics and Security*, 8(1):16–30.

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

Traoré, I., Nakkabi, Y., Saad, S., Sayed, B., Ardigo, J. D., and de Faria Quinan, P. M. (2017). Ensuring online exam integrity through continuous biometric authentication. In *Information Security Practices*, pages 73–81. Springer.

Zhong, Y., Deng, Y., and Jain, A. K. (2012). Keystroke dynamics for user authentication. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 117–123. IEEE.