

Continuous Authentication Methods for Zero-Trust Cybersecurity Architecture

Iurii Matiushin and Vladimir Korkhov(✉)

St Petersburg State University, 7-9 Universitetskaya Embankment, 199034 St Petersburg, Russia
v.korkhov@spbu.ru

Abstract. Zero-trust architecture (ZTA) is a new approach to distributed systems cybersecurity which both provides better data protection and reflects the structure of modern information systems more accurately. Unlike the traditional perimeter-style approach, ZTA views each part of the system as a potential threat, and continuously authenticates it to detect possible threats. This article presents a comparative analysis of the most common continuous authentication methods, and evaluates their viability for use in ZTA systems. Comparison criteria are introduced, most prominent authentication methods found in literature are listed, and their thorough evaluation is carried out. We conclude that, first, most continuous authentication approaches are not universal and can only be used with certain specific input methods; second, that face recognition, while a theoretically universal approach, has several issues of its own according to our criteria; third, that hybrid methods that combine several approaches are a potentially viable way of authentication in ZTA context; and fourth, that machine learning is essential to all of the present approaches.

Keywords: Cybersecurity · distributed systems · continuous authentication · zero-trust architecture

1 Introduction

1.1 The Need for a New Cybersecurity Approach

With each passing year, the issue of providing security for distributed systems is becoming more and more urgent. As the number of such systems and their complexity both increase, so does the amount of various cyberattacks on them, from within as well as from without. It is becoming increasingly clear that in many cases, most common security methods and paradigms are no longer sufficient for protecting one's systems and one's data [1]. Furthermore, the widespread shift to remote work, motivated in large part by the COVID-19 pandemic, has made its mark on the physical layout of many systems – no longer are their parts concentrated in a single location, instead being spread out among different buildings, cities, or even countries. Finally, the development of new technologies and approaches such as blockchain and IoT also call for novel methods of protecting the emerging systems. Thus, it becomes necessary to uncover new approaches to cybersecurity, ones that would provide protection against the latest threats and take into account the new realities of the world of computer systems.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
O. Gervasi et al. (Eds.): ICCSA 2023 Workshops, LNCS 14109, pp. 334–351, 2023.
https://doi.org/10.1007/978-3-031-37120-2_22

Zero-trust architecture (ZTA), also known as zero-trust security model, is an example of such an approach. While relatively recent, the concept of zero-trust security has quickly become very prominent, with 76% responders of a 2021 Microsoft poll among organizations claiming they have either implemented it or are interested in doing so [2]. Moreover, both companies and governments all over the world increasingly begin to consider ZTA to be one of the best choices when it comes to developing security strategy.

1.2 A Brief History of ZTA

To understand the meaning of zero-trust architecture, as well as the reason for its growing popularity, one should first consider the "traditional" approach to computer system security. This approach largely relies on a notion of a "security perimeter"; essentially, it treats the network as being divided into "outside" and "inside" zones. The outside zone – which is to say, the network outside the organization – is treated as inherently hostile and compromised, while the inside zone, being the part of the network that connects the organization's computational systems and data storage devices, is treated as safe. To separate the two, measures such as firewalls are used.

The approach described above is simple and intuitive, with clear separation of "safe zones" from potentially dangerous outside world. However, while it was the basis of computer system security for decades, certain recent developments have forced organizations to reconsider it.

Among such developments was the rise of concepts like cloud computing, Internet of Things, blockchain, and remote work policies. All these concepts meant that the perimeter itself was slowly disintegrating, as corporate systems grew more and more widespread and heterogeneous. In the past, connecting to a corporate network from the outside was the exception, whereas in recent years, it can often be the rule. Thus, drawing a line between the "inside zone" and the "outside zone" becomes harder and harder.

Another development was the recognition of an inherent vulnerability in the "security perimeter" concept. Namely, the fact that an entity that has managed to get past the perimeter security essentially has free reign inside the system. It effectively banks on the belief that the firewalls and other security systems can stop any threat from gaining access to the system; but if that belief proves to be incorrect, the entire system can become compromised as the result.

Starting from mid-2000s, security professionals began attempting to move away from the perimeter-based approach. In 2004, Paul Simmonds coined the term "deperimeterization", noting the need for a new data security model in his Jericho Forum presentation. In 2010, Forrester Research analyst John Kindervag popularized the term "zero trust" [3]. A year later, Google's zero-trust-based system called BeyondCorp was unveiled as a response to increasing cyberattacks; the company's articles on the subject also brought zero trust to the attention of many. In 2018, NIST released SP 800–207, the first specification for ZTA. By 2021, both US and British government cybersecurity organizations were recommending zero-trust approach for new IT deployments.

1.3 The Main Principles of ZTA

In the previous subsection, we described the traditional "security perimeter" approach, and indicated that zero-trust means abandoning it for something different. Essentially, if the traditional method can be summarized as "trust, but verify", then zero-trust can be summed up as "never trust, always verify". It basically does away with the division of network into "safe" and "unsafe" zones, and treats each and every entity as a potential threat. Thus, each interaction involves some sort of verification, in order to make sure that the entity has not been compromised since the last time.

In Fig. 1, the difference between the classic approach to system security and the zero-trust approach is shown [4].

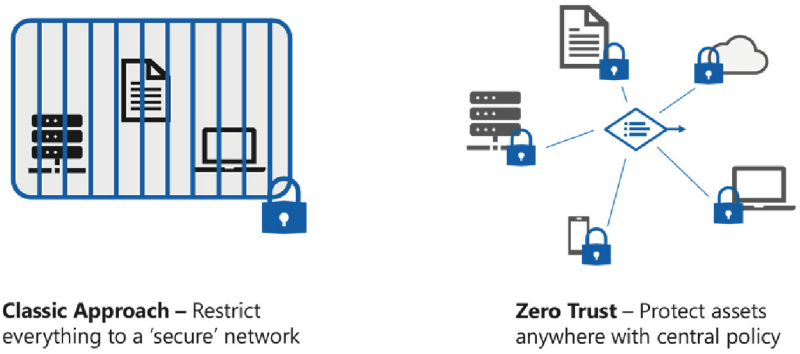

Fig. 1. Classic approach to security versus zero-trust approach.

Zero-trust architecture is built on three main principles: risk awareness, least privileged access, continuous authentication.

Risk awareness can be summarized as good knowledge of your system's vulnerabilities and possible threats. Security specialists need to be aware of all the assets that comprise the system – be it servers, personal computers, mobile devices, laptops, or anything else – and the ways users interact with them, all well as possible ways the security of each asset might be compromised.

The principle of least privilege means that each entity only gets as much access rights as necessary for its work and nothing more. This way, even if it is compromised, the possible damage is minimized.

Continuous authentication is a crucial part of zero-trust architecture. In a traditional perimeter-based model, authentication is carried out only once, at the beginning of the session; for ZTA, however, this is not acceptable. According to the "never trust, always verify" principle, the user's authenticity should be checked multiple times throughout the entire session, since it is possible that the user's credentials, or the device they are using to access the system, become compromised. So, the authentication mechanism must be capable of preventing unauthorized access not just once per session, but at any time.

This article examines various continuous authentication methods and determines their advantages and disadvantages as applied to their possible use in a zero-trust security

mechanism. We consider various factors that could influence a decision on whether to adopt any given method, and use them for a comprehensive comparative evaluation of different classes of methods. We also investigate the possibility of further development of certain methods, or using them in combination with one another.

The remainder of this paper is organized as follows. In Sect. 2, we provide an overview of relevant works on the topic of zero trust and continuous authentication. In Sect. 3, we introduce our comparison framework, listing the factors used for comparative analysis and explaining their significance. In Sect. 4, we enumerate the methods that will be compared, including their basic principles and existing implementations. In Sect. 5, the comparative analysis is carried out, and the methods are evaluated based on both quantifiable and non-quantifiable characteristics. Section 6 concludes the article, discusses the results of the previous sections, and lists possible directions for future work.

2 Literature Review

In this section, we investigate several articles that are relevant to our work, or have a similar topic.

In NIST Special Publication 800–207 "Zero Trust Architecture" [5], the standard for ZTA is described in detail. It details the requirements for a zero-trust security system, its necessary components, possible implementations, etc. Among other things, it also deals with authentication in such systems. In particular, it states that an authentication system must be strong, continuous, and yet unobtrusive to the user, lest it causes serious usability issues. The requirements set in the standard are used in this article to develop a comparison framework.

In "Zero Trust Architecture: Does It Help?" [6], E. Bertino discusses some challenges in implementing a zero-trust security system. In particular, it is stated that the zero-trust security policies must be correct, consistent, minimal, and complete; that is, they must align with the organization's goals, minimize and resolve permission conflicts, include no redundancies, and yet cover every possible situation when a user interacts with the system. Not fulfilling these requirements means that either the system's security, its usability, or both suffer; thus, they must be taken into account when designing a ZTA system.

In "A survey on continuous authentication methods in Internet of Things environment" [7], F. Hussain Al-Naji and R. Zagrouba provide a comprehensive analysis of continuous authentication algorithms that can be used in an IoT system. Their investigation is extensive and thorough, although centered mainly on the topics of IoT and using blockchain for authentication.

In "Security, Privacy, and Usability in Continuous Authentication: A Survey" [8], A. F. Baig and S. Eskeland present an overview of various continuous authentication methods, focused on three main factors – security, privacy, and usability. They discuss several methods, identify their issues and possible threat directions, and finally propose directions for future research. The authors list a number of possible approaches to authentication; they do not go into detail on each, preferring rather to provide a general overview and comparison.

In "Examining the Current Status and Emerging Trends in Continuous Authentication Technologies through Citation Network Analysis" [9], J. Jeong, Y. Zolotavkin, and R. Doss analyze the most frequently referenced works on the topic of continuous authentication; they use this analysis to investigate the current trends in this field. They discover that keystroke dynamics, mouse movement, and mobile device touch are three most popular approaches at the moment. Furthermore, they point out that there is a need for a consensus on how to evaluate the algorithms' application, utility, and feasibility.

In "Continuous authentication by free-text keystroke based on CNN and RNN" [10], X. Lu et al. propose a method for keystroke dynamics-based continuous authentication that uses neural networks to determine legitimate users. Unlike many other methods, the authors' algorithm is free-text, that is, it does not require the user to enter the exact same text (e. g. a password) to be authenticated. This makes it especially fitting for continuous authentication, since one of the goals is to be unobtrusive while keeping checking the user's authenticity throughout the entire session.

In "Continuous face authentication scheme for mobile devices with tracking and liveness detection" [11], M. Smith-Creasey, F. A. Albalooshi, and M. Rajarajan propose a continuous authentication algorithm based on face recognition. It is intended for use in mobile devices. The novelty is that instead of periodical re-authentication, the system tracks the user's face and only restarts when the face is lost. According to the authors, this approach is more secure, since it eliminates certain attack windows.

In "Continuous Authentication Using Mouse Movements, Machine Learning, and Minecraft" [12], N. Siddiqui, R. Dave, and N. Seliya discuss a dynamic biometrics continuous authentication method based on the user's mouse movement patterns. They note that the most common mouse movement dataset at the moment is too homogeneous and not representative of real-world applications. To gather heterogeneous data from the users for the authors' proposed dataset, a video game is utilized; machine learning, binary random forest in particular, is used to determine a user's authenticity.

In "Touch-based continuous mobile device authentication: State-of-the-art, challenges and opportunities" [13], A. Zaidi et al. provide a comprehensive review of authentication methods based on analyzing the user's touch behaviour pattern while using a mobile device. They propose it as a viable continuous authentication method, analogous in some ways to keystroke recognition; that said, the authors also recognize that certain challenges and obstacles still remain.

In "Continuous Authentication Based on User Interaction Behavior" [14], L. Chen et al. introduce a hybrid system that combines keystroke dynamic and mouse dynamic-based authentication methods. Furthermore, their system also takes into account the overall context and whether the user's interaction with the system deviates from the norm. This makes their work stand out from many other algorithms that only rely on biometrics, and provides a possibility to further look into combining various metrics in a continuous authentication system.

In "Secure Path: Block-Chaining IoT Information for Continuous Authentication in Smart Spaces" [15], L. Bracciale et al. propose another approach to user behaviour-based authentication. Their algorithm uses smart devices worn by users to determine their physical location and other relevant characteristics, which are then used by a distributed "trust committee" to determine if a user can be trusted.

In this section, we have reviewed various existing works on the topic of continuous authentication. The information about the most relevant qualities of such systems, as well as the most common approaches and their characteristics, will be used in the remainder of this article.

3 Method Comparison Framework

3.1 Introducing the Comparison Criteria

The methods we investigate in this article vary considerably. They are based on different principles and have different requirements; as such, meaningfully comparing them is no easy task. So, in order to carry out comparative analysis of various continuous authentication methods, suitable comparison criteria must first be identified.

In [6], Bertino discusses a similar topic, and introduces several requirements a ZTA system must meet in order to be useful. Based on this, as well as other considerations, we have decided to base our analysis on a list of factors that we consider to be essential for any authentication system.

The factors are as follows:

1. Accuracy
2. User convenience
3. Feasibility
4. Universality

Let us explore each criterion in more depth.

3.2 Accuracy

The importance of accuracy is obvious. Any authentication mechanism, in order to do its job effectively, must be good in telling the difference between the legitimate users and those who do not possess the necessary credentials, yet try to gain access to the system or its components anyway (whether with malicious intent or as mere error). The two most common parameters used to determine the authentication system accuracy are usually referred to as FRR and FAR.

FRR stands for False Rejection Rate. It refers to a percentage of times the system does not grant access to a legitimate user. It is also sometimes called a Type I error.

FAR stands for False Acceptance Rate. It refers to a percentage of times the system grants access to the user not possessing the necessary credentials. It is also sometimes called a Type II error.

Obviously, an ideal system will always grant access to legitimate users and will always deny it to everyone else. In the real world, however, neither FRR nor FAR can

be totally excluded from the equation. What determines an accurate system, then, is whether both FRR and FAR are low enough.

Another obvious consideration is that minimizing FRR and minimizing FAR are often somewhat opposite goals: a stricter system would likely have lower FAR and higher FRR, and vice versa for a more permissive one. We can imagine a system that always grants access to everyone and would thus have 0% FRR, or, on the other hand, a system that never grants access to anyone and would thus have 0% FAR. Both would, of course, be utterly useless in practice. Thus, it is necessary to find some balance between the two.

3.3 User Convenience

Unlike accuracy, user convenience is something that is often overlooked by security professionals, but is arguably just as vital. Any system must, in the end, serve its users, and if the security mechanism is too onerous for most users, it will seriously interfere with their experience. In a commercial context, it will likely lead to users and organizations abandoning the system altogether; in a work context, it will negatively affect the employee performance. Only when the data being protected is of utmost importance can the considerations for the user convenience be discarded.

Besides, low user convenience can lead to lapses in security as well. A simple example would be a company's employees (especially those not in IT or technical professions) being forced to memorize long and complicated passwords. Oftentimes, in practice the passwords are simply written down, which presents a serious security risk if a malicious party can gain physical access to the company's facilities.

For continuous authentication, user convenience is especially important. Using a secure yet somewhat onerous method like multi-factor authentication (MFA) once in the beginning of a work day would be acceptable to most people; however, the entire point of continuous authentication is that the entities in the system are authenticated much more often than that. Naturally, having to go through MFA many times per day is not something a lot of users would find acceptable.

3.4 Feasibility

By feasibility, we mean a number of factors that determine how easy and convenient an authentication method would be for an organization to implement. It includes various associated costs (for instance, if the method requires any special equipment to be installed), as well as how easy a method is to set up in an existing system.

Feasibility is another factor that should not be discarded from any analysis. For any organization, adopting a new technology is no easy task, and in many cases, it would not be done unless the organization understands the innovation, its importance, and how adopting it will be a benefit. In the field of authentication, passwords are a clear example – multiple security experts have proclaimed password-based authentication to be no longer effective, and plenty other methods have been developed; yet passwords remain the most popular method bar none. It is their simplicity and cheapness of implementation that contribute to them still being popular, as well as the fact that they are the "default option", and many organizations do not consider anything else.

Another example is continuous authentication in the IoT environment. As pointed out in [16], many IoT devices do not have the computational resources necessary for such a task; the authors suggest using blockchain technology to bypass this limitation.

Thus, in order for an authentication mechanism to be adopted in practice, it must be feasible for an organization to do so. Of course, like with user convenience, reasoning may differ depending on the nature of the organization and the data it is trying to protect – government entities would have a different set of goals and values compared to commercial ones. Still, the importance of feasibility should not be underestimated.

3.5 Universality

We also list universality as a criterion, by which we essentially mean whether an authentication method is platform independent. The inclusion of this factor is motivated by the fact that modern distributed systems are increasingly becoming more heterogeneous. Their components might include mobile devices, laptops, personal computers, servers, etc. So, for instance, if the authentication method only works for mobile phones, it will be a serious limitation in any other context.

3.6 Other Criteria

Aside from the four main criteria listed above, other miscellaneous factors and concerns may also be addressed if necessary.

3.7 Framework Summary

The framework for comparative analysis is summarized in Table 1.

Table 1. Authentication methods comparison framework.

No	Criterion	Meaning
1	Accuracy	FRR (false rejection rate) and FAR (false acceptance rate)
2	User convenience	Whether the method interferes with the users' normal experience
3	Feasibility	Possible complications that can arise during the method's implementation
4	Universality	Whether the method has any limitations regarding devices it can be used with
5	Other criteria	Any other concerns not addressed by the aforementioned sections

4 Continuous Authentication Methods

4.1 Overview

Continuous authentication can be implemented through use of various methods. The basic idea, however, remains the same: instead of authenticating a user once at the beginning of their session, the system keeps checking their authenticity throughout the entire session. The checking consists of comparing the user's current behaviour – as measured by some metric or algorithm – to their normal behaviour. If it is similar enough, the authentication algorithm considers the user to still be legitimate, otherwise, a more explicit check is used (e. g. MFA). In this way, continuous authentication works in tandem with existing user authentication methods, instead of replacing them completely.

Consider, for instance, an example where a user connects to the system through a mobile phone (e. g. opening a blockchain wallet application). If the user is successfully authenticated through a standard method, and then a third party gets their hands on the user's phone, this third party then gains access to the system for the duration of the session, and possibly indefinitely. Other similar cases include using keyloggers or social engineering, users connecting to unsafe third-party websites, etc. Continuous authentication helps prevent breaches in such instances – it is believed that a new entity using the existing entity's credentials will still behave in a sufficiently distinct way for the system to raise an alarm and investigate.

So, in order for the continuous authentication to work, it needs a way to distinguish one user's behaviour from another one's. There are multiple ways to achieve this; when it comes to human users, one of the most common approaches is to use static or dynamic biometrics.

Biometric authentication is based on a person's unique biological properties. The most common variety is static biometrics, including fingerprint scanning, facial recognition, retina scanning, etc. It analyzes some property of the user's body that is both unique and unchanging, converts it to a convenient data format, and compares it with the expected value of data. Dynamic biometrics, on the other hand, analyze some aspect of the user's behaviour – the way they type, move their mouse, walk, talk, etc.

Another way is to analyze the entity's actions in the system. For instance, if an entity starts requesting access to multiple files and locations it did not need before, sending or receiving larger-than-usual amounts of data, or performing other atypical actions, the zero-trust system can flag this as suspicious behaviour and act accordingly.

4.2 Keystroke-Based User Recognition

Keystroke biometrics is one of the oldest and most common types of dynamic biometric authentication. It is based on the principle that just as each person has their own distinct handwriting, the way different people type is also different.

There are many different features that can be used to establish a user's "signature". These include the time it takes to press or release a key, the time between key taps, the way a user enters certain common combinations of symbols (words like "the", "and", etc.), and many others.

The usual way to utilize keystroke recognition for user authentication purposes is to have the user enter some predetermined text. This can be used as a one-off security mechanism in the beginning of a session, but it does not particularly fit the requirements of continuous authentication. To authenticate the user throughout the session, it would require them to enter the same text multiple times at some time interval, which goes against our user convenience criterion. On the other hand, some of the newer algorithms use the so-called "free-text" keystroke recognition: the user does not have to enter any specific words or sentences, any text input will do.

Machine learning is a very useful tool for keystroke biometrics. A neural network can learn the keystroke patterns of a certain users, and then use them to verify if the input comes from a certain user or an impostor. For the free-text method, using neural networks seems to be the most efficient approach.

4.3 Face Identification

Face identification is one of the most common forms of biometric authentication. Unlike keystroke recognition, face identification is a type of static biometric algorithm, analyzing not user behavioral patterns, but rather some immutable characteristic unique to each user – in this case, the shape of their face and facial features.

Face recognition algorithms began their development in the 1960s, and have gradually grown more and more sophisticated and usable in real-world scenarios. Nowadays, they are used for an extremely wide variety of tasks, user authentication in various systems being one of the most common uses.

The main principle face recognition algorithms rely upon is that each human face has a large number of distinguishable characteristics, or landmarks: distance between the eyes, width of the nose, the shape of the cheekbones, etc. Taking these characteristics from an image and transforming them into digital format, an authentication algorithm gets some piece of data, which it then compares with the corresponding entry in its database. If it gets a fit – if the user's features correspond to those the one who currently claims to be the user possesses – then the authentication is successful.

4.4 Mouse Movement Recognition

Mouse movement recognition is another type of dynamic biometrics authentication algorithm. As the name suggests, it is based on recognizing unique patterns in the way a user moves their mouse and clicks the mouse buttons. It is more recent and less common compared to the algorithms described above; however, there is still a not-inconsiderable number of works on this topic.

Like with keystroke recognition, mouse movement recognition algorithms generally come in two types. One type deals with specific types of mouse movement, and gives the user certain tasks to fulfill in order to authenticate them. The other type uses general mouse movement, using the data generated by the users as they go about their daily activities on their computer. Once again, for continuous authentication purposes, the latter type is vastly more preferable.

Machine learning is as useful here as it is with other types of dynamic biometrics; in Sect. 5, ML-based mouse movement recognition algorithms are investigated.

4.5 Touch Gesture-based Authentication

Touch gesture-based authentication is yet another type of dynamic biometric authentication method that is currently gaining traction. The increasing popularity of mobile devices, as well as other types of equipment using touchscreens as an alternative or in addition to keyboard and mouse, makes this class of authentication algorithms more and more relevant.

Broadly, the similar principles apply here as with other user behaviour-based methods. In gesture-based authentication, the main point of interest is the swipe – a singular gesture made while interacting with a touchscreen. Such characteristics as the swipe's starting position, its trajectory, the average pressure, etc. can be taken into account. Some algorithms consider not just the individual swipes, but also the data from the entire session, like the average number of swipes and the average time between swipes. As with other algorithms, machine learning is the main tool utilized for confirming a user's authenticity based on their interaction with a touchscreen.

4.6 User Behaviour-Based Authentication

User behaviour-based authentication, unlike the majority of continuous authentication, algorithms, does not necessarily have to be based on biometrics. Rather, it analyzes the actions the user usually performs during a session, such as their interaction with various system components and applications, the requests they make from the system, etc., and then, based on this data, makes the decision on whether the user's actions in any given session are unusual and should be investigated further.

However, biometrics can also be a part of the authentication system, being one of the characteristics considered in order to determine whether the user behaves "as usual." So, some of the algorithms mentioned above can also be a part of a user behaviour-based method.

4.7 Authentication Methods Summary

The continuous authentication methods that we have described above are summarized in Table 2.

Table 2. Continuous authentication methods.

Method	Based on	Source
Keystroke-based	How the user presses the keys, enters certain combinations, etc	[10]
Face recognition	Continuous observation of the user's face via a camera	[11]
Mouse movement	How the user moves the mouse, clicks the mouse buttons, etc	[12]
Touch gesture	How the user interacts with a touchscreen device, presses, swipes, performs multi-touch gestures, etc	[13]
User behaviour	The patterns of the user's interactions with the system, possibly combined with other methods	[14]

5 Comparative Analysis

5.1 Methods Evaluation

Accuracy. Here, the accuracy of each method is considered, based on the results presented in existing literature.

Keystroke-based recognition's accuracy was evaluated in [10]. Depending on the dataset, FRR ranged from 1.89% to 6.61%, and FAR ranged from 2.83% to 5.31%.

Face identification's accuracy was evaluated in [11]. Depending on the dataset and the specific features used, FRR ranged from 1.49% to 37.89%, and FAR ranged from 1.05% to 60.81%.

Mouse movement recognition's accuracy was evaluated in [12]. A slightly different terms were used (false negatives rate and false positives rate), but overall, FRR ranged from 0% to 77.04%, and FAR ranged from 11.86% to 28.73%.

Touch gesture-based authentication's accuracy was evaluated in [13]. Multiple different approaches were mentioned, each with accuracy data from the relevant studies. Across different studies, FRR ranged from 1.65% to 28.52%, whereas FAR ranged from 0.04% to 22.45%.

User behaviour-based authentication's accuracy was evaluated in [14]. The authors claim FAR of 0% for all cases of their scenario, whereas FRR ranged from 0% to 5.88%. Other studies were mentioned as well, with maximum FRR of 2.10% and FAR of 5.7% across three different studies.

The distribution of FAR and FRR across different methods is shown in Fig. 2.

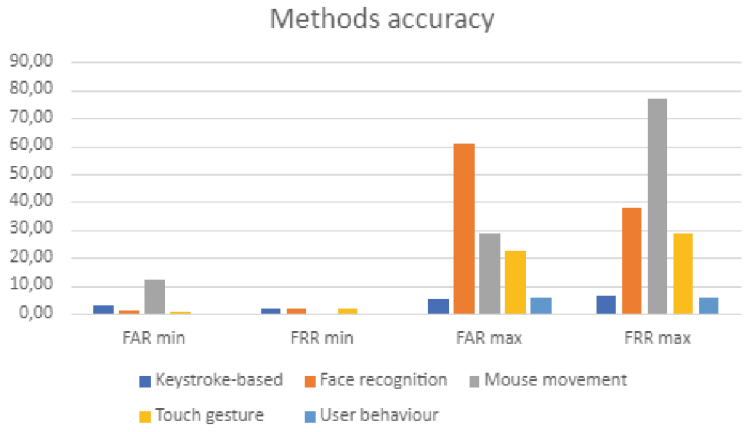

Fig. 2. Methods accuracy distribution.

User Convenience. Here, we investigate how convenient each method is for the end user, given that it will need to be used continuously throughout every session in the system.

Keystroke-based recognition is very convenient for the user if two conditions hold: first, that text-based input is used as the main method of interacting with the system (as

opposed to, for instance, mainly using a cursor), and second, that the recognition method is free-text: that is, *any* input text is acceptable to determine the user's authenticity. If the user does not use their keyboard while interacting with the system, it will need to prompt them to enter some text, which would interrupt the normal workflow. And, of course, if a user is forced to enter the same text several times throughout the session, there can be no convenience to speak of. However, if both of these conditions can be ensured, keystroke-based recognition would be one of the most unobtrusive methods to currently exist.

Face identification's convenience depends greatly on technology used. Some methods might not recognize the user's face from certain angles, or if the lighting is insufficient, or if a part of their face is obscured, or in some other cases. This is even more pronounced in mobile-based identification, where the device's camera has to be pointed at the user's face for the majority of the time. All in all, while this method cannot be called absolutely inconvenient per se, it still has certain inherent challenges the other approaches lack.

Mouse movement recognition is generally similar to keystroke-based recognition, with the necessary caveat that the user needs to chiefly use their cursor over their keyboard. Other than that, the same two conditions hold, and which method of the two is preferable depends on how an average user is projected to interact with the system.

Touch gesture-based authentication has the advantage that when it is used, there are generally no other alternatives, so there are less problems with users having to prioritize one method of input or the other. Devices with combined touch-based and keyboard/mouse-based input do exist, but they are less common. The other condition from above still holds – as long as the user does not have to fulfill a predetermined task, but can just be authenticated while working with the system, the convenience level will be sufficiently high.

User behaviour-based authentication is convenient if the user is expected to carry out similar actions during each session. If, however, the actions can differ in a significant manner, an alarm can be raised, even though the user is legitimate; this possibility can be considered as a negative for user convenience.

Feasibility This part looks into how cheap and simple the authentication method is to implement in an existing system.

Keystroke-based recognition has certain challenges when it comes to its practical implementation, particularly free-text recognition – which, as we have established earlier, is a must for a usable continuous authentication mechanism. Firstly, it is necessary to build up a considerable database of keyboard input samples for each user, so that it can be used to determine whether any given text input was typed by the user in question. Also, training the neural network – given that the most common approaches involve deep learning, CNNs, and RNNs – can take a non-inconsiderable amount of time and computational resources. So, the added requirements for data storage space and necessary computations should be taken into consideration, as well as the fact that the method needs to learn the user's "signature" before working properly.

Face identification requires that each device is outfitted with a video camera. This is not much of a problem for mobile devices, but can be an additional expense when it comes to desktop-based devices. If the distributed system is large enough, the financial

requirements would not be inconsiderable. Besides, a database of faces should also be created, though the mechanism to do so is older and more developed compared to other methods mentioned in this article, and the overall method would likely be more familiar to the majority of system owners and administrators.

Mouse movement recognition faces many of the similar problems to keystroke-based recognition. Namely, it needs to be free-form, so a database of mouse movement patterns must be built, and a neural network must be trained. The authors of one article [12] suggest a video game-based format to do so, but it might not be a good fit for every system.

Touch gesture-based authentication is also similar to keystroke-based recognition – in fact, many of the earlier methods were adapted from keystroke-based methods. That said, using a touchscreen adds certain peculiarities of its own; for instance, the fact that multi-touch is possible (unlike with a keyboard or a mouse) should also be taken into account.

User behaviour-based authentication faces many of the same issues other methods do, particularly if biometric algorithms are implemented as a part of it. In addition, a database of typical behavioural patterns for each user needs to be built up. For instance, in [14], the volunteers worked on data collection for 30 days before sufficient amount of user behaviour data was collected.

Universality. Here, we see how universal the method is – that is, whether it can be used with various types of devices that can be a part of a heterogeneous distributed system.

Keystroke-based recognition can be applied to systems that utilize a keyboard as an input mechanism. This is a rather broad category – including personal computers, laptops, mainframes, etc. – but it does exclude most mobile devices, which are becoming more and more common. Thus, this method cannot be called universal.

Face identification can be used with any device that has a camera. This includes most modern mobile devices, but it is less of a certainty when it comes to personal computers. Nevertheless, under the condition that a camera is either a part of the device a user is expected to use or can be installed separately, this authentication method can be used for a wide range of devices.

Mouse movement recognition broadly has the same limitations as keystroke-based recognition, since most modern devices that use a keyboard also use a mouse. It is also limited to such devices, and so is not universal.

Touch gesture-based authentication is, naturally, limited to devices that use a touch screen. As noted before, some of these devices also utilize a keyboard and a mouse; however, for such devices, touch screen is usually not the main input method, so a continuous authentication algorithm for them cannot be reasonably based on touch gesture recognition. This approach is also not universal.

User behaviour-based authentication's universality can vary, since this category is rather broad and can include other approaches in turn. In general, though, the idea of authenticating a user based on their behaviour in the system is not inherently limited to any one type of device or input method; therefore, this approach can generally be considered to be a more universal one.

Other Criteria. The one thing that is usually either absent from comparative analysis or is mentioned only in passing would be possible privacy concerns from the prospective

users. This is especially relevant since most methods we list involve biometrics – that is, the users will have to consent to having some of their features recorded, stored, and used for authentication, with the understanding that this data might be compromised in an attempt to gain access to the system.

We believe that face identification in particular can lead to such concerns. Face recognition technology is widely associated with biometrics, and, despite becoming more widespread in the recent years, has drawn both criticism and governmental restrictions in multiple countries [17]. Other methods discussed, on the other hand, are more obscure among the general public, and would likely lack such associations. That said, more in-depth research about public perception of different biometric authentication methods would be necessary to examine this issue in more detail.

5.2 Final Results

In this subsection, we sum up the results of our comparative analysis.

Overall, keystroke-based recognition was found to be the most consistently accurate approach, whereas for most other methods, both FRR and FAR can vary greatly, depending on the input data, the specific algorithms used, the usage scenarios, etc.

Most of the methods discussed above can be considered convenient for the end users, since they do not usually interrupt their normal workflow. Face recognition is likely the least convenient for the purposes of continuous authentication, since it needs the user to constantly look at the camera. User behaviour-based methods, meanwhile, need the user to interact with the system in a consistent, repetitive manner.

As for feasibility, most methods can be considered feasible to implement, since they do not require any additional hardware to be installed; face recognition is the exception, since it does need a camera for each device a user can utilize to connect to the system. All methods do, however, need to collect data on user input and/or behaviour patterns, and use it to train a neural network, which means additional storage and computational resources requirements.

Most of the methods cannot be considered universal, since they need specific input methods to work; once again, face recognition is an exception. Whether user behaviour-based authentication is universal depends on whether other biometric methods are a part of its implementation.

Finally, since most methods are biometric, possible privacy concerns should be taken into consideration, especially in case of face recognition.

The results are summarized in Table 3.

Table 3. Comparative analysis final results.

Method	Keystroke-based	Face recognition	Mouse movement-based	Touch gesture-based	User behaviour-based
FAR, %	2.83–5.31	1.05–60.81	11.86–28.73	0.04–22.45	0–5.7
FRR, %	1.89–6.61	1.49–37.89	0–77.04	1.65–28.52	0–5.88
Convenience	Convenient for free-text methods	Needs constant observation of the user	Convenient for free-input methods	Convenient for free-input methods	Needs stable user action patterns
Feasibility	Needs a neural network and a database of keystroke patterns	Needs a neural network + database + camera for each device	Needs a neural network and a database of mouse movement patterns	Needs a neural network and a database of touch gesture patterns	Needs a neural network + database for behaviour patterns and additional biometric methods
Universality	No, needs a keyboard	Possibly, needs a camera	No, needs a mouse	No, needs touchscreen	Possibly, depends on implementation
Other concerns	Minor privacy concerns	Major privacy concerns	Minor privacy concerns	Minor privacy concerns	Privacy concerns (if using biometrics)

6 Conclusions and Future Work

Continuous authentication is an essential part of zero-trust security – a new paradigm that can make for a much better protection of distributed systems in the age of blockchain, cloud computing, and remote work. In this article, we have carried out a comprehensive comparative analysis of five approaches to continuous user authentication in the context of a zero-trust security system – keystroke recognition, face recognition, mouse movement recognition, touch gesture recognition, and user behaviour analysis.

We have developed a framework for comparing different methods, and then used it to evaluate the most promising continuous authentication methods. Various factors were considered as a part of the framework – from recognition accuracy to user convenience and privacy concerns.

Most of the methods we have considered require some specific input method to work, so they cannot be universally applied to any devices in a heterogeneous system. Face recognition can be considered universal, but it has other limitations when it comes to feasibility, user convenience, and possible privacy concerns. User behaviour-based methods can be universal as well (depending on implementation specifics), and show good results in regard to authentication accuracy, but they need additional data about the user behaviour patterns, and besides, they require the user to interact with the system in a predictable manner.

As for accuracy, keystroke-based recognition and user behaviour-based methods show the best consistent results.

Also, as we have seen, there is no need to limit ourselves to using just one method. Hybrid approaches, combining two or more various metrics, are present in the literature. They allow the security system to become more universal – allowing it to be used for users with various devices with various input types – as well as more flexible, taking

into account multiple factors that help determine the user's authenticity. On the other hand, such systems can also be more complicated and difficult to implement compared to using a single metric.

Regarding the hybrid methods' accuracy, it can be estimated to be similar or higher compared to the methods using a single authentication factor. In particular, a method evaluated in this paper, combining user behaviour, keyboard, and mouse-based authentication, has demonstrated one of the best results in both FAR and FRR.

The field of continuous authentication is currently actively developing. New methods, or new variations of older methods – like free-text keystroke recognition, for instance – are being invented and put into practice. Of course, this means that many approaches are quite imperfect, and while they might not show the best results now, it is reasonable to expect notable improvements in the near future. Mouse movement recognition can serve as an example – the datasets existing now are for the most part limited and not representative of practical everyday use, but works are now appearing that seek to change that.

Machine learning is an essential tool for all methods of continuous authentication. As we have seen, every approach investigated is likely to involve neural networks in some way, shape, or form; the most common approach is to create a database of patterns associated with each user (ways of inputting data, face shape information, system interaction patterns) and then use it to determine whether the user is who they claim to be.

Based on the above, the following directions for future work present themselves:

- Further development of dynamic biometric methods suitable for use in zero-trust systems, i. e., continuous authentication methods. Examples include free-text keyboard recognition and non-specific mouse movement recognition.
- Discovering approaches in the field of machine learning most suitable for our task, based both on accuracy and other factors, such as feasibility and usability.
- Addressing the user privacy concerns, since most of the methods currently in use employ the users' biometric information. Depersonalizing the biometric data is one possible way to address these concerns.
- Considering hybrid continuous authentication methods.

Acknowledgment. This work was supported by Saint-Petersburg State University, project ID: 94062114.

References

1. Verizon 2022 data breach investigations report. https://www.verizon.com/business//resources/reports/dbir/2022/master-guide/. Accessed 02 Apr 2023
2. Microsoft security zero trust adoption report. https://query.prod.cms.rt.microsoft.com/cms/api/am/binary/RWGWha. Accessed 05 Apr 2023
3. Higgins, K.J.: Forrester pushes 'Zero trust' Model for security. https://www.darkreading.com/perimeter/forrester-pushes-zero-trust-model-for-security. Accessed 05 Apr 2023

4. Davis, J., Simos, M., Skoniecki, J.: Back to the future: what the Jericho forum taught us about modern security. https://www.microsoft.com/en-us/security/blog/2020/10/28/back-to-the-future-what-the-jericho-forum-taught-us-about-modern-security/. Accessed 05 Apr 2023
5. Rose, S., Borchert, O., Mitchell, S., Connelly, S.: Zero trust architecture. https://doi.org/10.6028/NIST.SP.800-207. Accessed 06 Apr 2023
6. Bertino, E.: Zero trust architecture: does it help? IEEE Secur. Priv. **5**(19), 95–96 (2021)
7. Hussain Al-Naji, F., Zagrouba, R.: A survey on continuous authentication methods in internet of things environment. Comput. Commun. **163**, 109–133 (2020)
8. Baig, A.F., Eskeland, S.: Security, privacy, and usability in continuous authentication: a survey. Sensors **21**(17), 59–67 (2021)
9. Jeong, J., Zolovatkin, Y., Doss, R.: Examining the current status and emerging trends in continuous authentication technologies through citation network analysis. ACM Comput. Surv. **56**(4), 31 (2022). Article 111
10. Lu, X., Zhang, S., Hui, P., Lio, P.: Continuous authentication by free-text keystroke based on CNN and RNN. Comput. Secur. **96**, 101861 (2020)
11. Smith-Creasey, M., Albalooshi, F.A., Rajarajan, M.: Continuous face authentication scheme for mobile devices with tracking and liveness detection. Microprocess. Microsyst. **63**, 147–157 (2018)
12. Siddiqui, N., Dave, R., Seliya, N.: Continuous authentication using mouse movements, machine learning, and Minecraft. In: Proceedings of the International Conference on Electrical, Computer and Energy Technologies (ICECET). IEEE, Cape Town, South Africa (2021)
13. Zaidi, A.Z., Chong, C.Y., Jin, Z., Parthiban, R., Sadiq, A.S.: Touch-based continuous mobile device authentication: state-of-the-art, challenges and opportunities. J. Netw. Comput. Appl. **191**, 103162 (2021)
14. Chen, L., Zhong, Y., Ai, W., Zhang, D.: Continuous authentication based on user interaction behavior. In: 7th International Symposium on Digital Forensics and Security (ISDFS), pp. 1–6. IEEE, Barcelos, Portugal (2019)
15. Bracciale, L., Loreti, P., Pisa, C., Shahidi, A.: Secure path: block-chaining IoT information for continuous authentication in smart spaces. IoT **2**(2), 326–340 (2021)
16. Hussain Al-Naji, F., Zagrouba, R.: CAB-IoT: continuous authentication architecture based on Blockchain for internet of things. J. King Saud Univ. Comput. Inf. Sci. **6**(34), 2497–2514 (2022)
17. Facial recognition technology and privacy concerns. https://www.isaca.org/resources/news-and-trends/newsletters/atisaca/2022/volume-51/facial-recognition-technology-and-privacy-concerns. Accessed 07 Apr 2023