June 2022

Brain-Dependent Biometric Authentication via Analysis of User Tasks, and Corresponding Cryptographic Key Generation

Will Drewry

BRAIN-DEPENDENT BIOMETRIC AUTHENTICATION VIA ANALYSIS OF USER TASKS, AND CORRESPONDING CRYPTOGRAPHIC KEY GENERATION

Abstract

Provided are computerized systems and methods for biometrically authenticating a user's identity using a learned representation of the user's neurological patterns, processes, and/or decision making, which may generally be referred to as a "biometric brain fingerprint". Example techniques include asking the user to perform a task and comparing their performance or response to a biometric brain fingerprint learned from prior user performance on tasks. For example, the user's response can be compared to a predicted response that has been predicted by a machine-learned biometric model that has been trained based on data from the prior user performance on tasks. The task can be nearly any task, including any task that involves application or stimulation of the user's executive function, such as an item selection task, sequence selection task, item manipulation tasks, and/or a simple captcha-like task. The machine learned biometric model can be implemented with lower complexity models such as a Markov model or can be done with more advanced techniques like recurrent neural networks, compact prediction trees, or support vector machines. In one example, a model can determine a probabilistic confidence level representing the model's confidence that the current user's performance matches a biometric brain fingerprint corresponding to the authentic user's performance of similar tasks. In another example, the user's performance can be compared (e.g., algorithmically or heuristically, such as with a distance measure) to a predicted performance predicted by the machine learning model. The proposed methods can be used in connection with existing authentication methods, such as two-factor identification and bootstrapping and/or can be used as part of an account recovery mechanism. The method can also be used as a seedable

challenge-and-response authentication framework, which can prevent spoofing attacks or replay attacks based on the recording or theft of biometric data. The task can also be used to create a cryptographic key specific to the user's brain fingerprint and/or specific to a particular transaction based on the challenge-and-response seed. The methods described in this paper also share some commonalities with, and may be combined with, methods from non-biometric cryptography, such as physical uncloneable functions and side channel analysis for side-channel attacks.

Background and Related Work

User Authentication

Many computing applications require user authentication, but many authentication methods have disadvantages. Two-factor authentication methods are often robust against theft, but may require a user to memorize a password. And because users often forget, most applications that use two-factor authentication also have a mechanism to allow users to recover their account after forgetting their username or password. These recovery mechanisms may be a point of vulnerability in a two-factor authentication scheme.

Biometric identification may avoid the need for a user to memorize information, but many forms of biometric identification can potentially be stolen in day-to-day life. For example, most people's eyes are visible whenever they go out in public; thus, a retina scanner placed in an unexpected location might capture biometric retina data from unsuspecting strangers. Thus, it would be useful to have a biometric marker that is stable, seedable, and cannot be easily captured in everyday interactions.

Neurological Studies of Biometric Brain Fingerprinting

Past research has established that a stable biometric brain fingerprint can be obtained through an EEG.[1] But an EEG scan is impractical for many mobile authentication applications. Another study has shown that a random number selection task can be used to uniquely identify users.[2] In that study, a simple statistical model was used to identify, with 88 percent accuracy, the user who created a particular sequence of numbers.

Cryptography

In cryptography, a challenge-response authentication is commonly used to prevent spoofing attacks by, e.g., a "man in the middle" who intercepts an encrypted communication between a user and a server. For example, in the Salted Challenge-Response Authentication Method (SCRAM), a server sends a "challenge" value to a client; the user then inputs a secret password into the client; and the client combines the password with the challenge value before encrypting the combined value and sending the encrypted information to the server. In this system, a system that intercepted the encrypted communication would be unable to determine the secret password, and would be unable to use the intercepted encrypted value to "spoof" an authentic user when the server sends a different challenge value. An even more secure form of challenge-response authentication is a physical unclonable function, in which a device's response to each challenge depends on unique, stable, and non-reproducible aspects of the physical structure of the device (e.g. microscopic imperfections unique to a particular silicon microchip).

[1] Maria V. Ruiz-Biondet et al, Permanence of the CEREBRE brain biometric protocol, https://www.semanticscholar.org/paper/Permanence-of-the-CEREBRE-brain-biometric-protocol-Ruiz-Blondet-Jin/3f530deb482ab2405fc2873b35577505b6190a0c

[2] Marc-Andre Schulz et al, Analysing Humanly Generated Random Number Sequences: A Pattern-Based Approach, https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0041531

Summary

A user's identity can be authenticated by asking the user to perform a task, and then comparing the user's performance to a known biometric fingerprint that represents the authentic user's performance predicted for the task and/or previously demonstrated on similar task(s). This can be done in many ways.

For example, in one implementation, an authentic user, when he or she first enrolls in a service or first creates an account, can be asked to perform a particular task, which can be almost any task because almost any task will be performed slightly differently by each user. For example, the user can be asked to perform a random selection task, such as a random number selection task, image selection task, color selection task, audio tone selection task, or physical input button selection task. For example, an initial series can be provided as a prompt and the user can be asked to choose the next item(s) in the series. The task given to the user can also be a goal-directed task, such as a simple puzzle-solving task, image rotation task, or captcha-like task. In some implementations, the task can merely request an input from the user, such that the micro-characteristics of the input may be measured. For example, an authentication server could ask a user to type a challenge sentence, which each user would type slightly differently. Measuring aspects of the user's typing behavior can create a biometric fingerprint of the user's typing behavior, which is directly linked with the executive function of the user's brain. In other examples, the task can be a passive task that requires relatively little input from the user, such as an image viewing task on a device capable of tracking a user's eye movement, such as a smartphone or other camera-equipped device. A mixed-activity task may also be chosen, e.g. a task that asks for number selection at some iterations; image rotation at other iterations; and passive tasks such as image viewing at other iterations.

However, in some cases, preferable choices for this task will be tasks that are accessible to all users, including those who may have impaired vision, impaired hearing, or a physical handicap. Thus, tasks that is usable with a wide variety of different accessibility devices (screen readers, voice-to-type devices, braille touch surfaces, etc.), such as a random number selection task, may in some cases be preferable to, for example, an image selection task that may be difficult for the visually impaired.

Similarly, challenge-response authentication can be used in a wide variety of contexts, including Internet-based services accessed phones, tablets, and PCs; secure element dongles (which are sometimes used as secure hardware wallets for crypto currency); wearable smart devices; and may more. Thus, preferable choices for a user task are ones that can be used across a wide variety of contexts, including a wide variety user devices and input mechanisms. However, in some cases, a specialized device-specific task may also be preferable, e.g. a task tailored to a specific device may be more convenient for the user in some cases. For example, a camera-equipped device capable of recording biometric eye-tracking data may be well suited for an image viewing task, which may be more convenient or more pleasant for a user than other, more device-agnostic tasks.

For a high-security application, a user can be asked to perform a task for many iterations, or to perform a very complex or time-consuming non-iterative task; for an application with less stringent security needs, fewer iterations or less user effort may be needed to authenticate users within an acceptable confidence threshold or risk tolerance threshold.

The task complexity or number of task iterations required to create a biometric fingerprint for a newly enrolling user can also be fixed or variable. For example, in some cases, an authentication server could ask a newly enrolling user to perform a fixed number of iterations,

such as 100, and then create a biometric fingerprint after the user is finished. But in other cases, an authentication server might monitor the user's performance on an ongoing basis, and ask the user to continue performing the task until the biometric fingerprint is detailed enough to meet the security needs of the particular application (e.g. its risk profile). For example, an authentication server could use well-known techniques from information theory to measure the information entropy of the input collected so far, and stop when an information entropy threshold is exceeded.

To create a biometric fingerprint based on the task performed, a wide variety of statistical or machine learning techniques can be used. For example, techniques as simple as a Markov model can be used to model many tasks. For example, a selection task can be analyzed with a sequence prediction model, e.g. a Markov model or compact prediction tree, to generate a stable biometric fingerprint of a user's selection sequences. More advanced statistical and machine learning techniques may also be used to create a biometric fingerprint from user task data. A statistical classifier or machine learning classifier can be used, e.g. a random forest classifier, AdaBoost Tree, Support Vector Machine, or other classifier. Many existing neural network or machine-learned models, e.g. a recurrent neural network, can be used to create a biometric fingerprint from user task data. Statistical and learning techniques from related fields can also be used, e.g. statistical techniques related to physically uncloneable functions or side-channel analysis. Future developments in neural networks and machine learning, e.g. improved machine-learning classifiers, will likely also be usable for creating a biometric fingerprint from user task data.

Once a biometric fingerprint has been created from user task data, the fingerprint can be used to authenticate a user, e.g. on future login attempts. At this stage, a user may be asked to

perform a short task, a simple task, or perform a set task for very few iterations for a low-security activity. In contrast, the user may be asked to perform a more complex or time-consuming task, or may be asked to perform more iterations of a fixed-complexity task, to authenticate before performing a higher-security activity. For example, a user who wants to click a "like" button may be permitted to do so without authentication. As another example, if a user wants to send $25 to a user who is known to be a "friend" of the sending user, he may be asked to perform two iterations of a random image selection task, with a 3x3 grid of images being shown at each iteration. As another example, if a user wants to make a $2,000 online purchase, with the purchased product being sent to an address not previously associated with the user, a larger number of iterations may be required to adequately authenticate a user.

The choice of task complexity or task iteration count may be based on a traditional risk profile and/or statistical confidence threshold. For example, a system may use a risk profile (e.g. an estimate of the risk associated with a desired user activity, or a risk categorization, or some other risk-associated value) to determine a required statistical confidence threshold for the authentication process. The system may then tailor the user task to ensure that the statistical confidence threshold is achieved. For example, an iterative user task, such as a color sequence selection task, may be repeatedly performed. A statistical confidence measure may be updated after each iteration of the iterative task, representing the system's confidence that the task iterations have been performed by the authentic user. If the current statistical confidence measure is below a target statistical confidence threshold, then the system may ask the user to continue performing more iterations of the iterative task.

Similarly, a statistical confidence threshold may be used to determine the required length or complexity of a non-iterative task, such as an image viewing task. For high-security

applications or high-risk activities, a user may be asked to view one or more images for a long period of time; for a lower-risk activity a user may only need to view an image briefly.

This task-based biometric fingerprinting method can also be incorporated into existing authentication systems, such as multi-factor authentication systems, bootstrapping, and/or cryptographic key generation. For example, a user may be asked to combine a fingerprint or iris scan with task-based fingerprint authentication. A multi-factor authentication like this may allow a statistical confidence threshold to be reached with less user input than a solely task-based authentication, while being less vulnerable to spoofing attacks than a solely iris-based authentication. The task-based system can also be used to generate a cryptographic key corresponding to the user's task-based biometric brain fingerprint. For example, with a complex enough task and/or a large enough number of task iterations, enough information entropy can be collected to generate a repeatable, stable, and unique key of a particular length. This key can then be used as a component of any traditional cryptographic protocol configured to use keys of that length. A challenge-response-based cryptographic protocol can also be specially adapted such that the challenge is embedded in the task itself, and the key generated is unique to both the authentic user and the challenge task.

Detailed Description

Figure 1 depicts an example computing system 100 on which the systems and methods disclosed here can be executed. The computing system comprises a user computing device 102 containing one or more processors 112, memory 114 which may contain data 116 and instructions 118 configured to carry out the methods disclosed herein, and a user input component 122. The user input component can be, for example, a touch screen or an input device attached to the user device, such as a mouse or keyboard. The computing system 100 further

comprises a network 180 and a server computing system 130. The server computing system 130 comprises one or more processors 132, and memory 134 which may contain data 136 and instructions 138 configured to carry out the methods disclosed herein. For example, a user may use the user input component 122 to input a money transfer request, e.g. a request to send $25 to a friend. The user device 102 may react to the input by sending a money transfer request through the network 180 to the server system 130. The server system 130 may then analyze the transfer request by retrieving instructions 138 from memory 134, such that the instructions are configured to cause the processor to analyze the money transfer request and generate a risk profile and/or security confidence threshold. To perform this analysis, the user may retrieve data 136, such as risk-related data, from memory 134. The server may then send a challenge task, such as a seeded random number selection task, to the user device over the network 180. The user may then perform the task using the user input component 122. After the user performs the task, the user device may send a representation of the user's task performance over the network 180 to the server 130, which then compares the representation to a task-based user fingerprint data, which is part of the data 136 stored in memory 134.

Figure 2 depicts an example authentication task when a preexisting user attempts to perform a user action on a user device, such as logging into an account using a new user device, which the user has never used to log into that account before. In response to the user's input, a user device may send a user request 210 over a network 180 to a server 130. Next, the server 130 may generate a risk profile 220 corresponding to a combination of the likelihood that the activity is fraudulent, the cost of allowing fraudulent activity (e.g. a money cost associated with permitting a fraudulent financial transaction, or an abstract representation of the cost associated with permitting fraudulent use of an email account to send spam), and other relevant factors.

Then, the server may choose a user task appropriate to that risk profile, or may choose a task without regard to risk profile, and may send the task as a challenge 230. The task may also be seeded such that the particular task is unique to the particular user activity request. Alternatively, for a low-risk activity, the user may choose not to send a challenge at all, and may grant the user activity request without authentication. The user may then perform the task 240, and the user device may send a response 250 comprising information associated with the user's performance of the task. The server may then perform an analysis and comparison 260 of the user's task performance compared to a stored measure of the user's past task performance, such as a biometric task fingerprint. If the analysis and comparison show that the user is authentic, with a confidence level exceeding a required confidence threshold, then the server 130 may grant the user request. If the confidence level is not exceeded, the server system may deny the request, or may send additional authentication challenges 230.

Figure 3 depicts an example biometric fingerprint generation method. First, a newly enrolling user may make an account creation request 310, which is sent from a user device 102 to a server 130. Next, the server system may send a challenge 320 asking a user to perform a task, which may optionally be seeded or unseeded. The user may then perform the task 330, and the user device 102 may data associated with the user's performance to the server 130 in a response 340. Next, the server may perform an analysis or processing step 350 on the data received in the response 340. The results of this can comprise a task-based biometric fingerprint, which can then be stored in the server's memory 134 in a storage step 360.

Figures

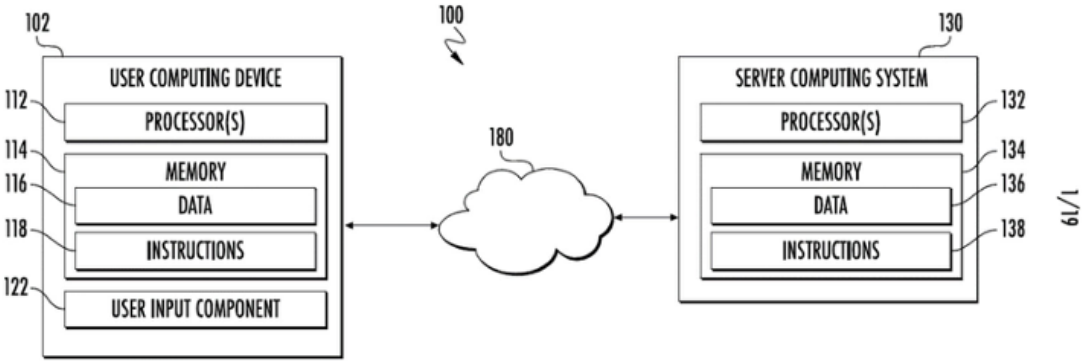

Fig. 1

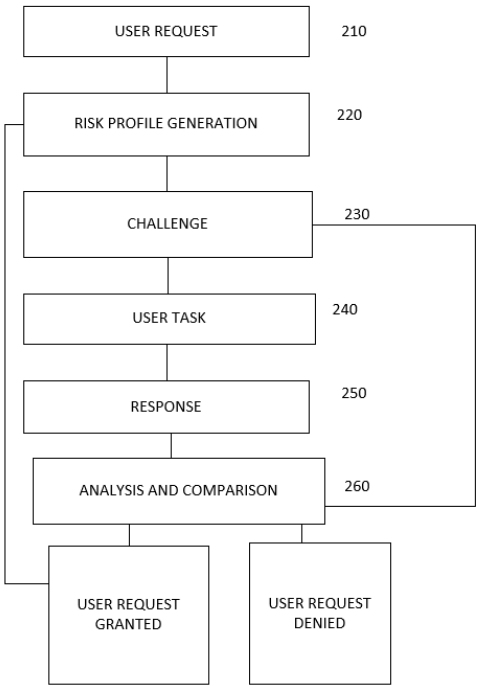

Fig. 2

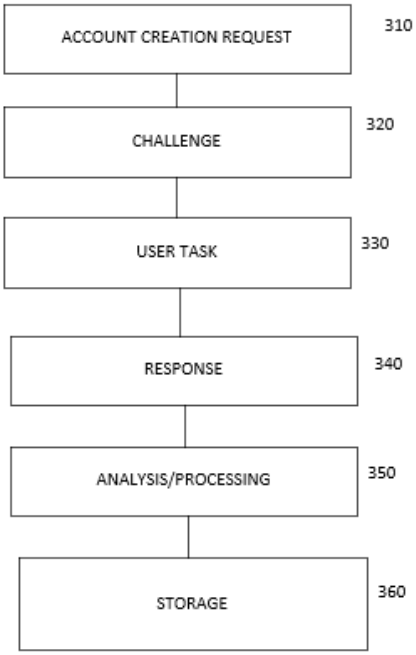

Fig. 3