Brain Computer Interface (BCI) Applications: Privacy Threats and Countermeasures

Hassan Takabi
Computer Science and Engineering
University of North Texas
Denton, TX, USA
Takabi@unt.edu

Anuj Bhalotiya
Computer Science and Engineering
University of North Texas
Denton, TX, USA
AnujBhalotiya@my.unt.edu

Manar Alohaly
Computer Science and Engineering
University of North Texas
Denton, TX, USA
ManarAlohaly@my.unt.edu

***Abstract*—In recent years, Brain-Computer Interfaces (BCIs) have gained popularity in non-medical domains such as the gaming, entertainment, personal health, and marketing industries. A growing number of companies offer various inexpensive consumer grade BCIs and some of these companies have recently introduced the concept of BCI "App stores" in order to facilitate the expansion of BCI applications and provide software development kits (SDKs) for other developers to create new applications for their devices. The BCI applications access to users' unique brainwave signals, which consequently allows them to make inferences about users' thoughts and mental processes. Since there are no specific standards that govern the development of BCI applications, its users are at the risk of privacy breaches. In this work, we perform first comprehensive analysis of BCI App stores including software development kits (SDKs), application programming interfaces (APIs), and BCI applications w.r.t privacy issues. The goal is to understand the way brainwave signals are handled by BCI applications and what threats to the privacy of users exist. Our findings show that most applications have unrestricted access to users' brainwave signals and can easily extract private information about their users without them even noticing. We discuss potential privacy threats posed by current practices used in BCI App stores and then describe some countermeasures that could be used to mitigate the privacy threats.**

I. Introduction

Brain-computer interfaces (BCIs) enable non-verbal communication between users and devices by using the measured brain signals for communication. BCIs have historically been used in medical domain to assist, augment or repair human cognitive or sensory-motor activities, and to enable people with paralysis to use their own brain signals to control assistive prosthetics like computers or robotic arms well enough to perform some everyday activities of living [1]. In recent years, however, BCIs have gained popularity in other domains such as the gaming, entertainment, personal health, and marketing industries.

A growing number of companies such as Emotiv [2], NeuroSky [3], MindSolutions [4], Muse [5], Melon [6], MyndPlay [7] offer various inexpensive consumer grade devices that are able to measure brain signals and record brain activities. More recently, OpenBCI offers a customizable and fully open brain computer interface platform, that can be used to sample EEG, ECG, EMG, and more [8]. MC10 is developing digital tattoo, flexible electronic circuits that stick directly to the skin like temporary tattoos and can measure EEG signals among other bio-signals [9]. Several marketing companies such as the Nielsen Company (who acquired NeuroFocus [10]) use EEG based devices for marketing research. In May 2015, Kokoon Technology released a Kickstarter campaign for Kokoon headphones, a headset that uses medical-grade EEG technology to actively support users to have better sleeping experience [11]. In the same year, the beta version of the first biometric social network was launched. The ongoing work on this project considers incorporating "EEG dating" idea with the social network as a feature for users of mobile BCI headset [12].

Some of these companies such as Emotiv [2] and NeuroSky [3] have recently introduced the concept of BCI "App stores" in order to facilitate the expansion of BCI applications. These App stores have a number of applications ranging from education, entertainment, gaming, meditation and relaxation training, etc. and they provide software development kits (SDK) for other developers to create new applications for their devices. As BCI technology spreads further towards becoming ubiquitous, it is expected future BCIs will be simpler to use and will require less training time and user effort, while enabling faster and more accurate decoding of users' intentions.

However, most of these applications have access to users' raw EEG signals and such unrestricted access to the raw EEG signals has raised several privacy concerns [1], [13], [14]. At the 2012 USENIX Security Symposium, Martinovic et al. introduced what they referred to as "brain spyware", the first BCI enabled malicious application designed to detect private information about a subject [13]. They used a commercially available BCI device, and developed an application which presents a subject with visual stimuli, then analyzes the recorded EEG signals to extract private information, such as the subject's: (a) 4-digit PIN, (b) bank information, (c) month of birth, (d) location of residence, and (e) if the subject recognized the presented set of faces.

978-1-5090-4607-2/16 $31.00 © 2016 IEEE
DOI 10.1109/CIC.2016.24

IEEE computer society

In a follow-up paper, Frank et al. identified a more serious threat for users of BCI devices [14]. They proposed a subliminal attack in which the victim is targeted at the levels below his cognitive perception and exposed to visual stimuli for a duration of 13.3 milliseconds, a duration usually not sufficient for conscious perception. The experimental results showed that it is feasible for an attacker to learn relevant private information about the user, and that if carried out carefully, the attack may remain undetected. As a result, the attack could be scaled to expose many victims to the attack for a long time [14]. Bonaci et al. studied how non-invasive BCI platforms used in games or web navigation, can be misused to extract users' private information [15]. They presented subliminal stimuli to the users for approximately 7ms while playing a game and logged their EEG signals to extract private information offline. For example, logos of retail stores or restaurants were presented as stimuli to determine if the user prefers Target over Walmart or McDonald's over Burger King and Kentucky Fried Chicken based on a person's reaction to those stimuli. In recent years, many investigators have shown how different components of EEG signals can be used to infer things about an individual's personality, memory, preferences, prejudices, religious and political beliefs, neurophysiological disorders, etc. For example, the P300 component of ERP has been used to recognize the subject's name in a random sequence of personal names [16], to discriminate familiar from unfamiliar faces [17], and for lie detection [18]. Similarly, the N400 component of ERP has been used to infer what a person is thinking about, after he/she is primed on a specific set of words. The P600 component of ERP has been used to make an inference about the person's sexual preferences [19], and an estimate of the anxiety level derived from the EEG signals was used to draw conclusions about the person's religious beliefs [20]. Seoane et al. introduce a proof of concept that uses BCI technology and EEG measurements to reconstruct a visual image which is on a user's mind [21].

In this paper, we report results of an analysis of the current BCI App stores and BCI apps w.r.t. privacy threats. This is the first comprehensive analysis of BCI App stores including BCI applications, SDKs, APIs. We also discuss potential privacy threats and countermeasures.

The rest of the paper is organized as follows. Section 2 describes the existing BCI app stores. In section 3, we describe the SDKs and the analysis of BCI apps. Section 4 discusses potential privacy threats to users of BCI apps and potential countermeasures. Related work is provided in Section 5, and finally, we conclude the paper in Section 6.

II. The BCI App Stores

A growing number of companies such as Emotiv [2], NeuroSky [3], Mind Solutions [4], OpenBCI [8], OCZ NIA [22], etc. offer BCI devices for consumer market. In this paper, we have mainly focused on two consumer grade BCI manufacturing companies that have "app stores" namely, NeuroSky [3] and Emotiv [2]. The goal is to provide a detailed analysis of the state of the app stores.

Table I: Number of Apps in App Stores

Platform ⟶		PC	Mac	Android	iOS
Emotiv	Free	8	4	-	-
	Paid	11	4	-	-
NeuroSky	Free	17	9	13	14
	Paid	58	38	15	15

A. NeuroSky Headset and App Store

There are various categories of applications such as gaming, education, wellness, research tools, and developer's tools in the NeuroSky App store. These applications run on different platforms including Android, iOS, PC (Windows), and Mac. NeuroSky headset, MindWave MOBILE, is a single dry sensor which sits on the user's left-forehead location with the reference electrode on the left earlobe [23]. It has the single channel, battery, and Bluetooth for connecting to the smartphone. The headset is embedded with a ThinkGear chip, which takes the raw EEG data picked by the single channel. It processes the EEG data internally by NeuroSky's proprietary algorithms and gives different data types, which are mentioned in next section. As shown in the Table I, as of June 30, 2016, there is total of 142 applications in NeuroSky app store, out of which 44 are free apps, and remaining 98 apps are paid.

For the app development with NeuroSky headset developers need to know about ThinkGear chip, ThinkGear API, ThinkGear Socket Protocol which are described in the following sections.

1) ThinkGear Chip: There are five different datatypes generated by ThinkGear chip embedded in MindWave Mobile EEG headset as described below:

Raw EEG Values (16 bit): First-byte value represents high order bits of 2' compliment value. Second-byte value represents low order bits. The default sampling rate is 512Hz. To change the sampling rate, then the developer must overwrite in the ThinkGear API.

Poor Signal Quality:This unsigned one-byte integer value describes how poor the signal measured by the ThinkGear is. It ranges from 0 – 255, and non-zero value represents noise. Integer value 200 shows that the device is not touching the skin.

eSense Meter: This has two categories, such as -
Attention eSense: Ranges from 1 – 100 values. Higher value represents strength of signal.
Meditation eSense: Ranges from 1 – 100 values. Higher value represents strength of signal.

EEG Band Power: This Data Value represents the current magnitude of 8 commonly-recognized types of EEG frequency bands (brainwaves). It consists of eight 4-byte floating point numbers in the following order:delta (0.5 - 2.75Hz), theta (3.5 - 6.75Hz), low-alpha (7.5 - 9.25Hz), high-alpha (10 - 11.75Hz), low-beta (13 - 16.75Hz), high-beta (18 - 29.75Hz), low-gamma (31 - 39.75Hz), and mid-gamma (41 - 49.75Hz).

Blink Strength: Ranges from 1 – 255 values. One byte value reports the intensity of the user's most recent eye blink.

2) ThinkGear API: The ThinkGear API first sets a connection with the MindWave Mobile headset and Android mobile application via Bluetooth. Once the channel is created, the ThinkGear API sends the required data, calculated on the ThinkGear chip, to the Android application.

Some of the gaming applications take the eSense meter values as input, and the developers should use the ThinkGear API to overwrite some classes to get more appropriate values of the eSense meter to enhance the gaming experience. Some of the methods applied to the eSense data are: convolve (), clone (), sum (), subtract (), max (), mean (), median (), smooth (), sort (), diff (), exceedValue (), square (), and std (). Finally, using these methods on the incoming data values from the headset, classification is performed.

3) ThinkGear Socket Protocol: The ThinkGear Socket Protocol (TGSP) is a JSON-based protocol for the transmission and receipt of the ThinkGear brainwaves data between a client and a server [24]. A server can be an application or a device that implements TGSP and authorizes requests and broadcasts the ThinkGear data. A client is an application or device that connects to a server. Stages in TGSP connection are: 1)*Creation of Socket*, 2)*Authorization*, given by the server to the client; 3)*Configuration*, where client can send commands to the server to configure transmission format or the data components; 4)*Receipt of data*, in which server will transmit data using streams,then the client uses JSON format to separate them using carriage return; and 5)*Termination*, during which the socket connection has to be terminated.

B. Emotiv Headset and App Store

Emotiv app store has several categories of applications including games, developer tools, and research tools which run on Mac OSX and Windows platforms. Emotiv headset, EPOC+, contains 14 non-invasive sensors which fit over the skull and 2 reference sensors, battery, and Bluetooth to have a connection with the smartphone. EPOC+ samples the EEG data at 128 or 256 per second. The datatypes of the Emotiv are described in the Emotiv SDK section III-A1. As shown in table I, as of June 30, 2016, there is total of 27 apps in the Emotiv app store, out of which 10 apps are free, and the remaining apps are paid.

For app development using the Emotiv EPOC+ headset, Emotiv API and its SDK are used which are briefly described in the followings.

1) Emotiv API: The Emotiv API [25] is developed on ANSI C interface, maintained in three header files (edk.h, EmoStateDll.h, and edkErrorCode.h), and implemented in 2 Windows DLL's (edk.dll and edk_utils.dll). *EmoEngine* is the driver for the EPOC+ headset - it is the DLL which includes all Emotiv detections, profile management, training systems etc. This is the engine behind the Control Panel application. It communicates with the *Emotiv headset*, receives preprocessed EEG and gyroscope data, and also manages post processing and translates the emo detection results into an easy-to-use structure called an *EmoState*. The Emotiv API functions used to modify or retrieve the *EmoEngine* settings are prefixed with *"IEE_."* An EmoState is a structure that contains the current state of the emo detections. The EmoState data is retrieved by the Emotiv API functions that are prefixed with *"IES_."* Handles are defined for *EmoState*. *IEE_EmoEngineEventCreate* is a method to create and *IEE_EmoEngineEventFree* is to deallocate the events of the EmoEngine respectively.

The abstract model for developing an application using Emotiv API is defined as: 1) Connection, 2) New event update, and 3) Disconnection. The application can connect to EmoEngine by calling *IEE_EngineConnect* if it wants to directly connect to the Emotiv EPOC headset. The API function *IEE_EngineRemoteConnect* will connect the application to a XavierComposer or an Emotiv Control Panel which are used as a simulator for getting the data for events. For terminating the application, users should explicitly close the connection with the EmoEngine by a *IEE_EngineDisconnect()*.

Main EmoEngine Events: There are three main types of events such as: Hardware-related event, New EmoState event, and Suite-Specific event. For Hardware-related events, either user is added in or out using **IEE_UserAdded**. For new EmoState events, either of **IEE_EmoStateUpdated** or **IEE_EmoEngineEventGetEmoState** can be used. For Suite-Specific events, **IEE_CognitivEvent** command can be used for training purpose.

EmoStateLogger: Logs all the Emotiv detections results, namely Expressiv, Assertiv, and Congnitiv.

Raw EEG Connection: The general approach for getting the data from the device is firstly connecting the EmoEngine to the headset, next creating a handle and adding a user, then finally starting to acquire the data by the following API function *IEE_DataAcquisitionEnable(userID,true)*. To retrieve the latest data from the buffer, the user calls the function *DataUpdateHandle()*.

III. Analysis of the Apps

In this section, we analyze SDKs of Emotiv and NeuroSky and the Android BCI applications from the NeuroSky app store, with the aim of gaining insight into how the EEG data is acquired, processed, and used. First, we talk about Emotiv SDK, then NeuroSky SDK, followed by applications from the NeuroSky app store.

A. Software Development Kits (SDKs)

1) Emotiv SDK: Emotiv SDK primarily consists of the neuroheadset, USB wireless receiver for pairing headset with a computer, and an installation CD. The analog brainwaves of the user are captured by the neuroheadset, then converted to digital form for further processing. The results are transmitted wirelessly to the USB receiver. Emotiv EmoEngine, a post-processing software, runs on the computer and reveals the results called Emotiv detection to the application via the Emotiv API [25].

For the purpose of analysis, we have installed the development kit on the Windows platform. The Emotive SDK contains three components: *Control Panel*, a program to test connectivity of the headset with the EmoEngine, which also helps in determining the headset sensors' quality and tunes the detection suites present in Emotiv SDK; *Documentation* like the API reference and the User Manual; and *Tools* like EmoComposer which emulate the EmoEngine and EmoKey which helps in mapping the EmoStates. The top bar on the Control Panel shows four indicators: a *System Status* signal; i.e. summary of the EmoEngine status; a *System Up Time* signal which shows the time (seconds) the EmoEngine is connected with the headset; a *Wireless Signal* indicates the strength of the connection between the headset and the Emotiv USB; and a *Battery Power* signal which displays the remaining charge in the neuroheadset. As mentioned above the Contol Panel assists in setting up the headset using a color scheme, while making sure that the headset sensors are fitted in place, so quality signals can be achieved.

The Control Panel has three types of suites which help in detecting the state of the users' brain: Expressiv, Affectiv, and Cognitiv [25].

Expressiv Suite: This Emotiv suite, now also known as Facial Expressions, represents an avatar which mimics the user's facial expressions like eye blinks, winks, brow movements, smiles, laughs, and clenches. It displays all these expressions in the form of graphs. This suite has two tabs. *Sensitivity*, when some of the user's facial expressions are not readily detected by the suite, then the sensitivity is increased or in case if some expressions are easily triggered or getting false positives, then the sensitivity is decreased for that expression. It could be increased or decreased by moving sensitivity slider to the right or left, respectively. *Training* tab gives a functionality where the user can train the system by performing the desired action before it can be detected.

Affectiv Suite: Affectiv suite, also known as Performance Metrics, shows real-time changes in subjects' emotions, which are Engagement, Instantaneous Excitement, and Long-Term Excitement. The brainwaves for these detections are universal in nature, so no training is required. For example, in a game, characters can transform in response to the player's feeling. Music, scene lighting, and effects can be tailored to heighten the experience for the user in real-time. The Affectiv suite can be used to monitor player's state of mind and allow developers to adjust the difficulty to suit each situation. This suite can be combined with other inputs like eye tracking devices to provide real-time feedback on the entire user experience for neuromarketing applications.

Cognitiv Suite: This suite, also known as Mental Commands, detects the user's real-time brainwaves activity concerning the conscious intent of performing different physical actions on a real or a virtual object. This suite can work with 13 different actions: 6 directional movements (push, pull, left, right, up, and down), 6 rotations (clockwise, counter-clockwise, left, right, forward, and backward), and one more that is disappearing. Cognitiv Suite allows the user to choose up to 4 actions that can be determined at a time. Cognitiv Suite also has a *Training* phase similar to Expressiv Suite, which allows the user to train the system and develop a personalized signature that corresponds to the particular action.

Tools: EmoKey and EmoComposer: EmoKey allows the user to connect the detection results gained from the EmoEngine to predefined keystrokes according to easy to use, logical rules. It has two parts: Rule definition and Trigger Condition. The rules are known as "EmoKey Mapping" and can be defined using the EmoKey user interface. The EmoKey communicates with the EmoEngine like a third party application does, using the Emotiv API. By default, if the Control Panel is running, then the EmoKey connects to it on its launch. The EmoKey has 3 fields for defining the rule: Name, Key, and Behaviour. Consider the following example: when a rule is defined for "laugh" in the rule mapping under the field *Name*, let's say the keyword 'LOL' is defined, then under *Key(s)* the keyword 'LOL' is defined again, and under the *Behavior* field checkbox for "Send Once" is ticked. Trigger Conditions are configured in order to define which corresponding rule should be activated for actions like Expressiv expression, Affectiv detection, or Cognitiv action. If above logical rule is defined for player1, an EmoKey mapping, and a trigger condition are configured for a third-party application, for example, Messenger. Considering the above rule, when a laugh event is triggered by Expressiv expression, the EmoKey will translate this event into text 'LOL'. As a result, whenever the user is laughing, the Messenger application automatically sends 'LOL'. EmoComposer, Emotiv's EmoEngine emulator tool, allows the user to send user-defined EmoStates to Emotiv Control Panel. It has two modes: Interactive and EmoScript. SDKLite users will rely on EmoComposer to simulate the behavior of Emotiv EmoEngine and Emotiv neuroheadsets.

2) NeuroSky SDK: NeuroSky's Mind Developer Tools (MDT or Developers Tools) contains a set of software tools that helps in developing BCI apps with NeuroSky device. MDT includes applications standards, Stream SDK, and EEG Algorithm SDK for Android.

Stream SDK for Android: Stream SDK helps to connect the application to the NeuroSky headset via Bluetooth and

Table II: DataTypes of NeuroSky Headset

Type	Description	Data
CODE_POOR_SIGNAL	init status	int
CODE_RAW	parsed raw data	int
CODE_ATTENTION	attention	int
CODE_MEDITATION	meditation	int
CODE_EEGPower	EEGPower	EEGPower object

receives bio-signal data from the headset. It has three parts as follows: TgStreamReader, TgStreamHandler, and MindDataType. *TgStreamReader* allows developers to manage the Bluetooth connection details and returns the parsed data. It includes four constructors used for reading raw signal data from a file, maintaining the Bluetooth connection between the app and the headset, and getting the stream and parsed data, connect to the dedicated Bluetooth device and help in getting data, and connect the headset with Android device via UART, and help in getting data. *TgStreamHandler* handles the callback messages and handlers are called in two situations: whenever data is received by calling the function *onDataReceived()* and when a state gets changed by calling function *onStateChanged()*. Finally, *MindDataType* outputs 5 types of data values as shown in the table II. The description of the datatypes are given in section II-A2.

The data received from the device is accessed using switch case statements. For example, if the app wants to access meditation value of the user, it can use the following snippet.

```
MindDataType.CODE_MEDITATION:
Log.d(TAG,HeadDataType.“CODE_MEDITATION”
    +msg.arg1);
tv_meditation.setText(“”+msg.arg1);
break;
```

SDK State defines different stages and how the control flows once the application is connected to NeuroSky headset.

EEG Algorithm SDK: The NeuroSky has proprietary algorithms for signal quality, meditation, attention, and raw data. The SDK assists the developers with the process of generating the algorithms' outputs from different NeuroSky Proprietary Mind Algorithm with EEG data collected by NeuroSky Biosensors. There are protocols defined for these four datatypes. The listeners which continuously listens are defined for these datatypes in the interval of 1 second to update values of the data types. Listeners for meditation are defined as *setOnMedAlgoListners*, for attention as *setOnAttAlgoListners*, and likewise for the other datatypes.

B. BCI Applications

In this section, we discuss results of our analysis of BCI app stores, specifically Android BCI applications from NeuroSky app store. Note that rather than Android framework in general, we focus on components that are added to the Android framework in order to handle BCI applications and how the EEG signals are acquired and processed by the applications.

The four core components in Android framework [26] that defines the app are Activities, Services, Content providers, and Broadcast receivers. An activity represents a single screen with a user interface, for example, first activity represents BCI app connecting to BCI device via Bluetooth. The Second activity might be for showing the graph of user's brain signals and third for showing options for storing the raw EEG signals and etc. *Service* component runs in the background to perform long-running operations or remote process. For example, a BCI app can use service to play music as stimuli for collecting quality meditation signal values. *Content provider* manages a shared set of app data for purposes like storing data (EEG signals) to SD card, cloud, etc. Also, it is useful for reading and writing data that is private to user's app as some of the BCI applications requires user's previous data for dynamic analysis. Finally,*Broadcast receiver* component responds to system-wide broadcast announcements used by the system to broadcast message like low battery, the screen turned off, etc.

The activity, service, and broadcast receiver components are activated by an asynchronous message called an *intent*. Activity can be activated by providing intent to *startActivity()* or *startActivityForResult()*. Service can be activated by passing intent to *startService()* or can be bound using *bindService()*. Similarly, Broadcast can be activated by passing an intent to methods like *sendBroadcast()*, *sendOrderBroadcast()* or *sendStickyBroadcast()*. The manifest file in Android is known as *AndroidManifest.XML*, present in the root directory of the app, and provides essential information about the components in the app to the Android system.

In order to analyze the BCI applications, we get the Android application package (APKs) which contains source code, necessary data files, and resource files from the smartphone directory location “\data\app”. Then, the .apk file is decompiled using a package of tools that contains *dex2jar*, *jd-gui-Windows*, and *apktool* [27]. By running .apk file on dex2jar, we get the classes-dex2jar.jar file and by running jar file on jd-gui-Windows, it outputs the packages containing the source code of the application. Finally, by running the same .apk file on apktool, we get AndroidManifest.XML and packages such as assets, res, etc.

There are many free android analysis tools available such as Dexter [28], Infer [29], Lint [30], Qark [31]. We first used Dexter where one uploads the .apk file and gets the analysis report which contains the permissions needed by the application, list of java classes, and a package dependency graph. The dependency graph represents the abstract view of the control flow of an application which is useful for our purpose. However, Dexter is not able to analyze the large .apk files. Likewise, the other available tools don't give the satisfactory result for our analysis purpose. As a result, we

Table III: BCI Applications and their Permissions

Application	Bluetooth	Storage	Photo/Media Files	Identity	Contacts	Location	Phone	In-app Purchases	WiFi Info	SMS	Camera	Device & App history
Biozen	✔	✔	✔	✔	✔	✔	✔	×	✔	✔	×	✔
Brain Visualizer	✔	×	×	×	×	×	×	×	×	×	×	×
Cardgazer	✔	✔	✔	✔	✔	×	✔	✔	×	×	×	×
EEG Analyzer	✔	✔	✔	×	×	×	×	×	×	×	×	×
eegID	✔	✔	✔	×	×	✔	×	×	×	×	×	×
Mind Jukebox	✔	✔	✔	✔	✔	✔	×	×	×	×	×	×
NeuroFun	✔	×	×	×	×	×	×	×	×	×	×	×
Puzzlebox Orbit	✔	×	×	×	×	×	×	×	×	×	×	×
Blink Camera	✔	✔	✔	×	×	×	×	×	×	×	✔	×
BrainPrint	✔	×	×	×	×	×	×	×	×	×	×	×
EEG Profile	✔	×	×	×	×	×	×	×	×	×	×	×
FlappyMind	✔	×	×	×	×	×	×	×	×	×	×	×
Frogs EveryWear	✔	✔	✔	×	×	×	×	✔	×	×	×	×
MyndPlayer	✔	✔	✔	✔	✔	×	×	✔	×	×	×	×
Rorschach	✔	×	×	×	×	×	×	×	×	×	×	×
ThinkTalk	✔	×	×	×	×	×	×	×	×	×	×	×
Transcend	✔	×	×	×	×	×	×	×	×	×	×	×
Transcend 2	✔	×	×	×	×	×	×	×	×	×	×	×
UpCake	✔	×	×	×	×	×	×	×	×	×	×	×
UpCake 2.0	✔	✔	✔	×	×	×	×	×	×	×	×	×

opted for manual code analysis.

We did the manual code analysis by following the flow in the manifest.XML file. Our approach for manual code analysis was to start with the java file (source code) associated with the first <activity>declaration in the app manifest file. When activity launches for the very first time, *onCreate()* in that activity, is the entry point for Android, just like main() is the entry point in C, C++, or Java programming. Likewise, onStart(), onResume(), and onRestart() are the callback methods used for activity to set the activity up and onStop(), onPause(), and onDestroy() are the callback methods used to put hold or tear it down completely. We have followed every intent message passing from one of the above-mentioned callback methods which lead to other packages or java files. We analyzed the classes, methods, and the parameters that are being passed and the returned values of the corresponding methods. The general pattern for almost every BCI app we analyzed is as follows. At first, welcome screen shows up that is managed by one activity, then the application connects to the headset via Bluetooth and checks for signal quality that is taken care of by other activity class. Once the connection is established and return an integer value for the signal quality metric is received in the range of 0-2 , the stimulus which may be a video, picture, or music is started by other activity, and the headset starts sending EEG data to the application.

As mentioned in the Table I, NeuroSky app store holds 28 android applications out of which 13 are free apps and remaining 15 are paid apps. We investigated 10 free apps and 10 paid apps as some of the apps were not available for the US store. Out of the total 20 analyzed apps, 9 are gaming apps, 4 are education apps, and 7 are wellness apps. Since any android or iOS app requests for uses permissions on installation, our very first checkpoint is to study "uses permissions" required by the BCI app. As shown in Table III, almost all the BCI apps inspected ask for the permissions such as Bluetooth permission, SD card access permission, Wifi connection permission, and camera access permission for apps needing a camera. Surprisingly, some third party BCI apps ask for access to phone book, call logs access permissions which should not be required for the BCI apps.

ThinkGear data values can be accessed using either ThinkGear API or ThinkGear socket protocol. The ThinkGear data is in the form of streams which contains all the data values mentioned in II. For using individual datatype, developers have to parse the data from the streams. Many third-party application developers use frameworks for graphics, virtual currency for the apps in conjunction with ThinkGear API or ThinkGear socket protocol.

We grouped all the inspected apps into two main categories: the ones that use the ThinkGear API and the ones that use the ThinkGear socket protocol. Our findings suggest that 16 apps used the former while the remaining 4 apps used the latter. Further, we divided the ThinkGear API used apps according to the datatypes (values) those apps receive from the API. The subsets consist of following: sampling rate at 128 Hz or 512 Hz, Signal Poor Quality(SPQ), Blink Strength(BS), eSense Meter (meditation and attention), Band

Table IV: Data Values Accessed by BCI Apps (TGAPI: ThinkGear API, SPQ: Signal Poor Quality, BS: Blink Strength, eSense: Meditation and Attention values, BP: Band Power, FFT: Fast Fourier Transform, DWT: Discrete Wavelet Transform)

Application	ThinkGear Data Values					App's Own Algorithm		Cloud Storage
	TGAPI	SPQ	BS	eSense	BP	FFT	DWT	
Biozen	×	✔	✔	✔	✔	✔	✔	×
Brain Visualizer	✔	✔	✔	✔	✔	×	×	×
Cardgazer	✔	✔	✔	✔	×	×	×	×
EEG Analyzer	✔	✔	✔	✔	✔	✔	×	✔
eegID	✔	✔	✔	✔	✔	×	×	×
Mind Jukebox	×	✔	✔	✔	✔	×	×	×
NeuroFun	✔	✔	✔	✔	×	×	×	×
Puzzlebox Orbit	✔	✔	✔	✔	×	×	×	×
Blink Camera	✔	✔	✔	✔	×	×	×	×
BrainPrint	✔	✔	✔	✔	✔	×	×	×
EEG Profile	×	✔	✔	✔	×	×	×	×
FlappyMind	✔	✔	✔	✔	×	×	×	×
Frogs EveryWear	×	✔	✔	✔	×	×	×	×
MyndPlayer	✔	✔	✔	✔	×	×	×	×
Rorschach	✔	✔	✔	✔	×	×	×	×
ThinkTalk	✔	✔	✔	✔	×	×	×	×
Transcend	✔	✔	✔	✔	✔	×	×	×
Transcend 2	✔	✔	✔	✔	×	×	×	×
UpCake	✔	✔	✔	✔	×	×	×	×
UpCake2.0	✔	✔	✔	✔	×	×	×	×

Powers(BP) (alpha, beta, gamma, delta, and theta), Raw EEG Data, Data Processing methods, and Classification of processed data. Similarly, we also split the set of apps which use the ThinkGear socket protocol into two groups: the ones that use the ThinkGear datatype values and the ones that implement their own developed algorithms (FFT, WLT, and DWT) for processing raw EEG data.

We found out that 5 apps use all the ThinkGear data values, and 11 apps fetch values like SPQ, BS, and eSense Meter using the ThinkGear API. The set of the apps which use the ThinkGear socket protocol, access the majority of the ThinkGear datatypes as shown in Table IV. These apps also process raw EEG data using different algorithms such as Fast Fourier transform [32], Wavelet transform [33], and Discrete wavelet transform [34] and apply classification algorithms to the data. In general, all the apps have unrestricted access to raw EEG signals and eSense Meter values. We also found out that a third-party application sends data to the cloud. Although we did not find any malicious intent in any of those apps, but as demonstrated in [35] the BCI game developers can collect the meditation and attention level values of the individuals while they are playing the game and later use some machine learning techniques to infer users' preferences which is a privacy threat.

IV. Privacy Threats and Countermeasures

In this section, we first discuss privacy repercussions to the users of the BCI applications. Then, we discuss potential countermeasures to preserve the privacy of users' brainwaves signals while enabling the BCI applications to use the signals for their legitimate purpose.

A. Privacy Threats of BCI Applications

The third-party BCI application developer can get unrestricted access to raw EEG signals through the APIs. Moreover, the app developers have complete discretion over the stimuli that can be shown to the user. The malicious developers could also mount any stimuli (video, image, etc.) they want to the application and collect users' signals in reaction to those stimuli. These will allow them to analyze the raw signals and extract private information about the users. Other issue is that the applications can send the raw EEG data to external servers and later the attacker or the developer apply machine learning techniques on that raw data to extract private information about the user. As shown in Table IV, some applications send raw EEG data to cloud storage.

There are many situations where EEG data is collected and released publicly for research/ clinical purposes. In order to protect the privacy of owners of EEG data, typically the datasets are anonymized and personally identifiable information (PII) are removed before release. However, as it has been shown, anonymization alone is not enough and linking attacks could be used along with other publicly available datasets to potentially identify owners of the EEG data. This is especially exacerbated with the availability of public genome data which could lead to the identification of users and exposure of their health data. Exposing of health data to malicious attacker can also put the users' life at risk.

The another critical issue is that our ability to extract meaningful information from brain signals is rapidly growing, so it is not yet known the extent to which sensitive information may be extracted from neural data in the future.

For example, if one's EEG signals are collected for research purposes, will they be identified from EEG databases that will be available in future? Or, will the researchers of tomorrow be able to learn their potential neurophysiological disorders? In light of the potential of growing neuroscience field in extracting private information from EEG signals in near future, privacy issues should be taken into account seriously before distribution of sensitive data.

Finally, the potential consequence of privacy breaches can be far worse than a breach in the traditional applications because we are trying to protect an individual's free will, health, ability to think, and cognitive liberty instead of protecting the software on their computers.

B. Countermeasures

The main issue with BCI App stores is that current APIs available to third-party developers offer full access to the raw brainwave signals and the users of the application have no control on to their brainwaves. The possible ways of restricting access to raw EEG signals are as follows:

Access Control: The raw brainwave signals from the BCI devices should not be given directly to the applications. One way to restrict access to the raw EEG signals is to design and develop an ecosystem similar to that of mobile applications where we will have a context-aware access control system that determines if requests from third-party applications are legitimate and makes decisions about them. The decisions should be made about what entities should have access to an individuals neural data; should those entities have access only to certain components of the neural data, certain features of the neural data, data from certain parts of the brain, etc.; what risks are associated with the misuse of those components; which purposes are the entities allowed to use the neural signals for; under what conditions should the entities be allowed to present individuals with new stimuli, etc.

Anonymization Techniques: Another potential solution is to remove private information from raw EEG signals before they are stored and transmitted as suggested in [36]. Although this can prevent some information leakage, it only works when feature extraction algorithms of any BCI-enabled app are fully known. However, the applications often develop proprietary algorithms and they would not like to reveal them.

The heart of BCI technologies is the ability to process EEG data in order to extract meaningful information. Similar to signal processing in general, processing EEG signals contain three main phases: pre-processing, feature extraction and features interpretation. In order to protect privacy of users' EEG data in various steps of processing, we can develop techniques based on modern cryptographic techniques such as homomorphic encryption [37] and functional encryption [38].

Functional Encryption: If the only goal is to preserve the privacy of the raw EEG signals, approaches based on functional encryption are more appropriate as the feature extraction phase should be done in real time so approaches based on homomorphic encryption are not practical although they work in theory. In functional encryption, decryption enables one to learn only a specific function of the data, unlike public-key encryption, where one can access to either all message or nothing. Functional encryption is completely non-interactive, once one obtains the functional secret key, he can do the decryption and obtain intended information. Applying functional encryption on BCI apps to protect the privacy of EEG signals could be considered as a potential solution.

Homomorphic Encryption: If the goal is to preserve the privacy of extracted features in addition to the raw signals, approaches based on homomorphic encryption are a better fit. Once the features are extracted from raw EEG signals, different methods such as machine learning algorithms are applied to the feature vectors to interpret them into meaningful results. Traditionally, machine learning algorithms require access to the raw data (feature vectors in our case) and in BCI applications, it is important that the feature vector remains secret. Ideally, the BCI application should not be able to learn anything about the user's input. The goal here is to provide solutions to run machine learning algorithms on encrypted EEG data and allow the BCI applications to run the algorithms and process the EEG data without having the data owner to reveal their sensitive EEG data to the applications. Recent advances in fully homomorphic encryption (FHE) enable a limited set of operations to be performed on encrypted data but more research is needed to develop practical efficient approaches for this purpose.

Randomization and differential privacy [39] have been used to distort time-series data of multiple users to preserve the utility of aggregate results. Applying randomization techniques on an individual time-series data that is not aggregated over many samples should be investigated in future because they are more applicable for real-time applications compared to cryptographic techniques. Furthermore, adding noise to the EEG raw data before making it available to the applications could be another solution. As suggested in [40], differential privacy can be used and implemented on the BCI hardware or software in order to get anonymized brainwaves signals of the users which prevent users' private information being extracted from the signals. However, finding the trade-off between privacy protection and utility of the signals is an important issue that needs to be investigated further.

V. Related Work

The National Academies of Sciences, Engineering, and Medicine's Committee on Science, Technology, and Law (CSTL) discussed possible legal and ethical issues raised by neural engineering focusing on the privacy concerns with respect to mental functioning [41]. Although the need for addressing emerging privacy issues in growing EEG-based applications has been acknowledged [42], [43], [44], [41], the practical solutions for preserving privacy are missing. Bonaci et al. identify security and privacy issues of BCI systems and argue that to address those issues, we need an interdisciplinary approach by experts from different areas, such as neuroscientists, neural engineers, ethicists, as well as legal, security and privacy experts [1]. They also propose "BCI Anonymizer" to prevent the extraction of users' private information [43]. Its basic idea is to remove private information from raw EEG signals before they are stored and transmitted. While this is a good idea, it suffers from several shortcomings. Firstly, the idea remains at the concept level and the authors do not provide any details on how to support the concept, how private information is identified in raw signals, what algorithms are used, how it is developed and implemented in practice, etc. Second, the authors suggest that the BCI Anonymizer could be implemented either in hardware or in software, as a part of BCI devices, but not as part of any external network or computational platform. However, BCI devices are typically resource constrained and may not have the processing capability to perform the required computations on a huge amount of signals that is collected. Third, the BCI anonymizer can protect privacy only when feature extraction algorithms of any BCI-enabled app are fully known. However, the applications often develop proprietary algorithms and they would not like to reveal them. Fourth, the authors do not provide a clear method to distinguish between command related information and user's private information.

Stopczynski et al. propose an integration of a personal neuroinformatics system, Smartphone Brain Scanner, with a general privacy framework openPDS [44]. In this framework, raw high dimensional data is collected on a mobile device, sent to a trusted server to be processed and then features are accessed by applications. In this approach, the server needs to know all the feature extraction algorithms. Furthermore, it requires a trusted server and adds huge communication burden which is problematic for scenarios such as BCI applications which need real time processing. Both of the above approaches can only protect raw signals while our proposed research can also protect the extracted features that could be sensitive themselves. Li et al. discuss the security and privacy challenges of brain-computer interfaces with respect to BCI applications [45]. They classify BCI applications into four scenarios: 1) neuromedical applications, 2) user authentication, 3) gaming and entertainment, and 4) smartphone-based applications, and discuss security and privacy issues for each scenario.

VI. Conclusion and Future Work

BCI technology is becoming ubiquitous and a growing number of companies offer inexpensive BCI devices for consumers and some of them have introduced App stores to encourage more developers to build BCI applications. However, most of these applications have unrestricted access to users' raw EEG signals which has raised privacy concerns. In this paper, we investigated the NeuroSky and Emotiv App stores including the applications, APIs and SDKs in order to evaluate the privacy issues of BCI applications. Our findings show that all applications are capable of collecting EEG signals of their users and extracting private information about them. We then discussed potential privacy threats of BCI applications and provided countermeasures to mitigate these threats. For future work, we plan to implement some of the discussed countermeasures for the existing BCI App stores.

References

[1] T. Bonaci, R. Calo, and H. J. Chizeck, "App stores for the brain: Privacy & security in brain-computer interfaces," in *Ethics in Science, Technology and Engineering, 2014 IEEE International Symposium on*. IEEE, 2014, pp. 1–7.

[2] Brainwear®wireless eeg technology. Emotiv. (Accessed June. 30, 2016). [Online]. Available: www.emotiv.com/

[3] Eeg-ecg-biosensors. NeuroSky. (Accessed June. 30, 2016). [Online]. Available: www.neurosky.com/

[4] Mindsolutions. (Accessed July. 26, 2016). [Online]. Available: www.mindsolutionscorp.com

[5] Muse. (Accessed July. 26, 2016). [Online]. Available: www.choosemuse.com

[6] Melon. (Accessed July. 26, 2016). [Online]. Available: www.thinkmelon.com

[7] Myndplay. (Accessed July. 26, 2016). [Online]. Available: www.myndplay.com

[8] Openbci r&d kit. (Accessed July. 26, 2016). [Online]. Available: www.openbci.com

[9] Mc10 inc. (Accessed July. 26, 2016). [Online]. Available: www.mc10inc.com

[10] Nielsen company. (Accessed July. 26, 2016). [Online]. Available: www.nielsen.com/us/en/solutions/capabilities/consumer-neuroscience.html

[11] Kokoon eeg headphones. (Accessed July 18, 2016). [Online]. Available: www.neurogadget.net/2015/06/24/kokoon-eeg-headphones-monitor-brainwaves-promise-better-sleep/11455

[12] First biometric social network. (Accessed July. 18, 2016). [Online]. Available: www.neurogadget.com/2015/06/17/first-biometric-social-network/11417

[13] I. Martinovic, D. Davies, M. Frank, D. Perito, T. Ros, and D. Song, "On the feasibility of side-channel attacks with brain-computer interfaces," in *Presented as part of the 21st USENIX Security Symposium (USENIX Security 12)*, 2012, pp. 143–158.

[14] M. Frank, T. Hwu, S. Jain, R. Knight, I. Martinovic, P. Mittal, D. Perito, and D. Song, "Subliminal probing for private information via eeg-based bci devices," *arXiv preprint arXiv:1312.6052*, 2013.

[15] T. L. B. M. T. Bonaci, J. Herron and H. J. Chizeck, "How susceptible is the brain to the side-channel private information extraction," *American Journal of Bioethics, Neuroscience*, vol. vol. 6, no. 4, 2015.

[16] J. P. Rosenfeld, J. R. Biroschak, and J. J. Furedy, "P300-based detection of concealed autobiographical versus incidentally acquired information in target and non-target paradigms," *International Journal of Psychophysiology*, vol. 60, no. 3, pp. 251–259, 2006.

[17] S. Marcel and J. d. R. Millán, "Person authentication using brainwaves (eeg) and maximum a posteriori model adaptation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 4, pp. 743–752, 2007.

[18] V. Abootalebi, M. H. Moradi, and M. A. Khalilzadeh, "A new approach for eeg feature extraction in p300-based lie detection," *Computer methods and programs in biomedicine*, vol. 94, no. 1, pp. 48–57, 2009.

[19] C. Mühl, B. Reuderink, M. Poel, Y. Yao, R. Sun, T. Poggio, J. Liu, N. Zhong, J. Huang *et al.*, "Guessing what's on your mind: Using the n400 in brain computer interfaces," 2010.

[20] M. Inzlicht, I. McGregor, J. B. Hirsh, and K. Nash, "Neural markers of religious conviction," *Psychological Science*, vol. 20, no. 3, pp. 385–392, 2009.

[21] L. F. Seoane, S. Gabler, and B. Blankertz, "Images from the mind: Bci image evolution based on rapid serial visual presentation of polygon primitives," *Brain-Computer Interfaces*, vol. 2, no. 1, pp. 40–56, 2015.

[22] Ocz's neural impulse actuator. (Accessed July. 26, 2016). [Online]. Available: www.guru3d.com/articles-pages/ocz-nia-review-neural-impulse-actuator,1.html

[23] T. Yamauchi, K. Xiao, C. Bowman, and A. Mueen, "Dynamic time warping: A single dry electrode eeg study in a self-paced learning task," in *Affective Computing and Intelligent Interaction (ACII), 2015 International Conference on*. IEEE, 2015, pp. 56–62.

[24] Neurosky socket protocol. (Accessed June. 30, 2016). [Online]. Available: www.developer.neurosky.com/docs/lib/exe/fetch.php?media=-app_notes:thinkgear_socket_protocol.pdf

[25] Emotiv(sdk)- user manual. (Accessed July. 3, 2016). [Online]. Available: www.github.com/Emotiv/community-sdk

[26] Android app components. (Accessed June. 15, 2016). [Online]. Available: www.developer.android.com/guide/components/intents-filters.html

[27] Process for reverse engineering android app. (Accessed June. 30, 2016). [Online]. Available: www.stackoverflow.com/questions/12732882/reverse-engineering-from-an-apk-file-to-a-project

[28] Dexter-static analysis tool. (Accessed June. 30, 2016). [Online]. Available: www.dexter.dexlabs.org

[29] Infer:static analyzer tool. (Accessed August. 6, 2016). [Online]. Available: www.fbinfer.com/

[30] Lint: Static analyzer tool. (Accessed August. 6, 2016). [Online]. Available: www.developer.android.com/studio/write/lint.html/

[31] Qark:quick android review kit. (Accessed August. 6, 2016). [Online]. Available: www.android-arsenal.com/details/1/2470

[32] Fast-fourier-transform. (Accessed August. 6, 2016). [Online]. Available: https://en.wikipedia.org/wiki/fast_fourier_transform

[33] Wavelet transform. (Accessed August. 6, 2016). [Online]. Available: https://en.wikipedia.org/wiki/Wavelet_transform

[34] Discrete wavelet transform. (Accessed August. 6, 2016). [Online]. Available: https://en.wikipedia.org/wiki/Discrete_wavelet_transform

[35] J.-Y. Kim and W. Lee, "Eeg signal feature analysis of smartphone game user," *Advanced Science and Technology Letters*, vol. 39, pp. 14–19, 2013.

[36] T. Bonaci and H. J. Chizeck, "1bonaci, chizeck1@ ee. washington. edu," 2013.

[37] C. Gentry, "A fully homomorphic encryption scheme [ph. d. thesis]," *International Journal of Distributed Sensor Networks, Stanford University*, 2009.

[38] D. Boneh, A. Sahai, and B. Waters, "Functional encryption: Definitions and challenges," in *Theory of Cryptography Conference*. Springer, 2011, pp. 253–273.

[39] C. Dwork, "Differential privacy: A survey of results," in *International Conference on Theory and Applications of Models of Computation*. Springer, 2008, pp. 1–19.

[40] A. D. Sarwate and K. Chaudhuri, "Signal processing and machine learning with differential privacy: Algorithms and challenges for continuous data," *IEEE signal processing magazine*, vol. 30, no. 5, pp. 86–94, 2013.

[41] N. Y. C. B. Association *et al.*, "Are your thoughts your own," *Neuroprivacy and the legal implications of brain imaging. New York: The Committee on Science and Law*, 2005.

[42] N. A. Farahany, "Incriminating thoughts," *Stanford Law Review*, vol. 64, p. 351, 2012.

[43] H. J. Chizeck and T. Bonaci, "Brain-computer interface anonymizer," Feb. 6 2014, uS Patent App. 14/174,818.

[44] A. Stopczynski, D. Greenwood, L. K. Hansen, and A. Pentland, "Privacy for personal neuroinformatics," *Available at SSRN 2427564*, 2014.

[45] Q. Li, D. Ding, and M. Conti, "Brain-computer interface applications: Security and privacy challenges," in *Communications and Network Security (CNS), 2015 IEEE Conference on*. IEEE, 2015, pp. 663–666.