MOBILE USER AUTHENTICATION AND THE PROMISE OF GRIP GESTURES

Phani Soumya Inguva[#1], Urban Jaklin[#2], Krishna Sindhuja Kalusani[#3], Christian Merchant[#4]
Department of Computer Science & Engineering, University of South Carolina
[1]pinguva@email.sc.edu
[2]jakline@email.sc.edu
[3]kalusani@email.sc.edu
[4]merchane@email.sc.edu

Abstract

The ubiquity of smartphones has seen a comparative rise in security concerns related to these devices. In particular, security concerns with mobile user authentication have been much addressed in the literature as researchers have realized the potential for information leakage and the threat it poses. In this paper, we trace the rise of smartphone security in the marketplace and describe current problems with user authentication. We focus on the study of grip gestures for the purpose of user authentication and propose a novel framework for how grip gesture, Bluetooth, and pressure sensors may be applied to mobile user authentication in the automotive domain.

I. Introduction

A. Current Concerns with Mobile Devices

The steep increase of mobile device usage in recent years has shown a similar increase in the realm of mobile device malware and security vulnerabilities. With people using their personal mobile devices for tasks as menial as checking social media or as important and confidential as checking their card transaction histories and account balances, there is reasonable cause for concern with regard to the security of devices such as smartphones and tablets. The major problems within the mobile domain can be seen on devices implementing the Android mobile operating system. Since Android holds a majority of the user base of smartphones and tablets, it is noted that this dominion in the market share is mirrored in the reception of mobile malware. Approximately 97% of all mobile malware is targeting Android, with new threats nearly quadrupling each year. This occurrence is compared in parallel to the increased usage of the Windows operating system. As more standard computer users chose Windows as their operating system of choice, more malware was created to target this system in particular. It is believed the same thing is happening with Android today--since more standard users are choosing Android, hackers are creating more malicious payloads and malware to target devices running this operating system directly [20].

Compounding upon these general concerns with Android devices, there are specific vulnerabilities which have been uncovered recently which further reveal the levels of insecurity on these mobile platforms. As an example, it has been shown in research published in 2013 that pre-installed “bloatware” on Android devices (implemented by vendors and service providers) can cause approximately 60% of Android devices’ vulnerability issues. Since the basic system format and layout of Android devices allows for extensive customization, vendors often take advantage of this to supply users with applications and software on the devices which is often impossible to remove. This “bloatware” is able to have over-privileged access to crucial components within the operating system and firmware. Vendors often make significant changes to the phone’s firmware, creating even larger areas of vulnerability [31].

The case of imperfections with core components leading to non-secure systems is not a rare incident in mobile devices’ recent history. A security researcher at mobile security firm Zimperium claimed in fall of 2015 that over 95% of Android phones can be hacked with an MMS message. This vulnerability was found in a core Android component dubbed Stagefright. This component is used to process, play, and record multimedia files--a very common activity for the typical smartphone user. The Zimperium researcher notes that a user doesn’t even need to open the MMS message to receive the malware--simply receiving the message will hack the device. Once this vulnerability was found, a solution was developed and sent to Google along with a report of the problem. Google was able to apply the patches to its internal code base within 48 hours, but since the turnover of Android updates is so slow, most Android phones could still be hacked using this method for quite some time until the patch was applied across all Android mobile platforms [6].

The main contribution of this paper is a novel, theoretical model for applying a set of current technologies and form factors—Bluetooth, pressure sensors, and steering wheels—to the task of authenticating smartphone users in automotive contexts. Since most concerns with mobile devices discussed so far were related to hacking and the reception of malicious payloads, there is another area of mobile device security which faces significant security threats and is more closely related to the paper’s major contribution—user authentication.

II. Related Work

A. Problems with Physical Gesture in User Authentication

Problems with user authentication in the mobile domain abound. Often this is due to the fact that traditional multi-factor authentication methods are too inefficient for use in the mobile domain. Mobile users prefer usability to security and vendors cater to the demands of the marketplace [2]. Intuitively, this behavior is clearly understood as users unlock their smartphone potentially hundreds of times per day. A cumbersome but secure process of authentication is absurdly impractical. As a result of this weakness, novel methods for extracting user information by exploiting security vulnerabilities continue to be exposed every year. In 2016, for instance, researchers in Singapore were able to sniff user data while a smartphone was being charged at a mobile charging station in a variation of the juice film attack [15].

A discussion of every vulnerability in user authentication is beyond the scope of this paper, and thus the Android operating system is our focus as a result of its dominance in the global consumer marketplace [29]. Specifically, in light of our novel work in kinetic gesture for the purpose of user authentication, graphical passwords are analyzed in detail so that the reader may understand the level of vulnerability that is possible given this popular authentication method when coupled with Android's dominance.

All user-knowledge authentication schemes (recall authentication) are susceptible to guessing—like dictionary and naive brute force attacks [5]. The idea of graphical passwords was proposed as an alternate authentication channel to dictionary attacks and other text based methodologies. In particular, the widespread use of Google's Android operating system has propagated graphical authentication because of its widespread use around the world [13]. The figure below shows Android's 3x3 graphical authentication grid:

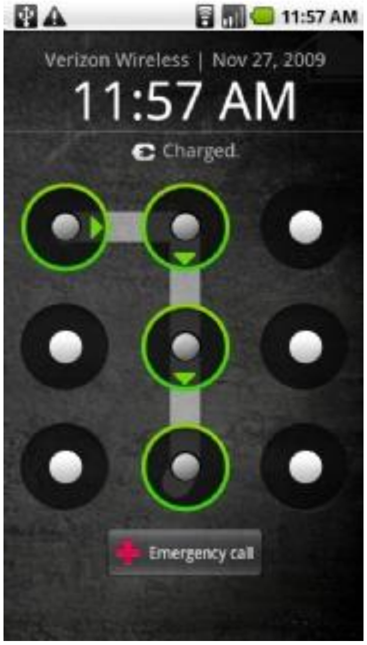

Figure 1, Android's 3x3 graphical authentication grid [4]

Users are free to create their own unlock pattern from the figure, provided they adhere to certain constraints on their ordered sequence of contact points:

1. The pattern must contain at least four points, so single strokes are not permitted
2. If a contact point is an element of the pattern it may be used exactly once
3. If there exists a contact point between two other contact points, it must also be a contact point.

These three constraints effectively mean that the pattern will have, at minimum, one direction change. Perhaps most importantly, the limitations on unlock patterns also decrease the sample space of all Android password combinations without these restrictions from almost 1,000,000 to 389, 112 [3]. This decreased sample space can aid attackers in selecting likely pattern combinations users would employ. In 2010, researchers at the University of Pennsylvania discovered that, far from being more secure than text-based passwords, Android graphical passwords are actually vulnerable to smudge-based attacks [3]. Smudges are the oily residue that is transferred from the pads of fingers to the smartphone screen.

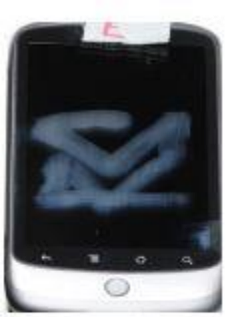

Figure 2, Showing smudge capture on Android screen [3]

This is especially applicable to Android's graphical password entry as user enters their password by sliding a finger around to each of the contact points on the screen, leaving behind a pattern-based smudge. The researchers, using cameras available in the marketplace as well as multiple lighting positions, explored whether these smudges could be extracted from the phone in an an identifiable way [3]. They describe their findings for extraction of patterns:

> *Our results are extremely encouraging: in one experiment, the pattern was partially identifiable in 92% and fully in 68% of the tested lighting and camera setups. Even in our worst performing experiment, under less than ideal pattern entry conditions, the pattern can be partially extracted in 37% of the setups and fully in 14% of them* [3].

At the time of this writing, there is little in the literature that directly addresses this problem in an elegant way. It can be postulated that manufacturers view user convenience as a primary concern over sound security measures. Cumbersome techniques for verification do not appear to gain market traction in mobile applications.

Human beings are exceptional at pattern recognition [14]. It logically follows that in creation of their own graphical unlock patterns, low-entropy themes emerge. Uellenbeck and others discovered that in 20% of cases studied, Android graphical patterns had slightly lower entropy than a three digit assigned PIN number [28]. Further, in 50% of cases, patterns could be mapped back to a sample space of fewer than 300 templates; 10% of the population uses fewer than 190 patterns [28]. Bias in human selection is obvious in the study of initial points of contact for users in pattern creation:

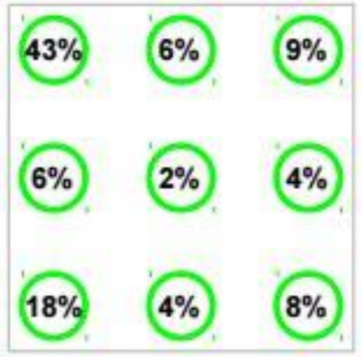

Figure 3, Initial point bias in sampling of 584 participants [27]

The result of this bias is apparent in the ability of researchers to guess the passwords of users in 4% of cases where the phone locks after ten attempts [28]. Usability plays an enhanced part in this process, as locking a user out of their phone after only a few failed guesses is poor business practice.

Shoulder surfing research is not a recent area of interest in the literature. Many have attempted to solve the problem of users directly observed or recorded with cameras entering their passwords [1, 30, 10]. The problem with these and other graphical authentication schemes is that they often compromise efficiency or were shown to be vulnerable to certain types of attacks [22, 25]. Going back to usability, users want to log in to their smartphone conveniently. Any scheme that inhibits this process even marginally will not be adopted by vendors because of its effect on user experience. Problems with Android graphical patterns were surveyed because of the widespread use. It is clear that if security is to be strengthened, the process must be intuitive and seamless. The user should not be aware that it is happening and yet it must be robust enough to provide stronger security. The use of pressure sensors in a smartphone is discussed below. One could combine this with text or graphical based approaches that are processed simultaneously. The parallel processing of kinetic and recall-based authentication would achieve multi-factor authentication without the user being aware that it is happening. Similarly, it would be more challenging for an attacker to spoof both login procedures.

B. Pressure Sensors and Grip Gestures

Due to the drawbacks of physical authentication techniques involving physical, behavioral and psychological patterns, one might say there is a need for "grip gesture" authentication. This is a natural technique involving no complex procedures or a need for memorizing things. The only action the user has to perform for unlocking the phone is gripping the phone.

Figure 4, Unlocking a phone using grip technique [9]

Focusing on the working of the proposed technique, the user is authenticated based on the pressure distribution of their individual grip gesture using the pressure sensors mounted on the lateral and back of a mobile phone and along the steering wheel in the case of cars. A successful authentication is achieved in five phases described as follows:

1) *Acquisition of Pressure Data*: Initially, a profile is set up from the values obtained using the pressure sensors. This profile data is defined a training set. Every time the user attempts authentication for a successful login, the system extracts feature values from raw pressure data obtained from the sensors. The system then compares the extracted feature values with the training data.
2) *Calculation of Time Difference*: The pressure values are captured by the pressure sensors mounted on the device. As the pressure sensors might be soft and easy to deform, there are some mathematical procedures included to calculate the time difference and to obtain values that are not distorted.
3) *Gesture Spotting*: Gesture spotting plays a pivotal role in the process of grip gesture authentication as the grip gesture values must be obtained from the stream of pressure data. This phase obtains the required accuracy by defining proper threshold values.
4) *Feature Extraction and Distance Calculation*: Feature values are extracted to calculate distances between training data and input data. Grip position, grip timing and grip force are the three essential elements involved in obtaining the pressure values.
5) *Authentication Decision*: The system calculates the distance for all the training data and chooses the smallest of all. This smallest value is then compared to the previously defined threshold value. If the distance value is smaller than the threshold value, the system unlocks the phone. Else, a retry is requested. [18]

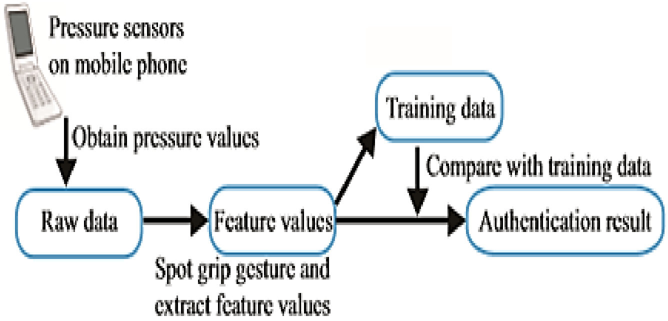

Figure 5, Flowchart showing the working of grip gestures [16]

This procedure is simple, secure and precise with error rate found equaling that of the facial recognition technique from previous works.

Applications:

1. Human eye movement tracking, grip gestures and posture analysis could be combined together to help in performing psychological operations (psyops) to identify user's intention. With the rise of terrorist activity today, this could prove very helpful. [16]

Figure 6, Psyops [16]

2. In attendance-based courses where some students would prefer bunking classes and Professors would like to keep a watch on those students, it is difficult as someone could sign up the attendance for the missing classmates. Grip gesture based attendance plays a perfect solution to this kind of a scenario.

3. Burglary is random and unsafe or low level security in unlocking the home doors could be the prime reason for this. Unlocking the doors using grip gesture might turn out to be a good solution for this problem.
4. Useful in designs which support grip gestures to control living room features such as lighting. Worldkit system is one application of grip gestures designed for this [23].

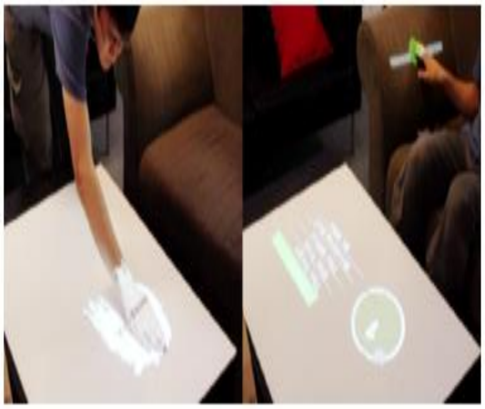

Figure 7, "WorldKit system: a user performs a swipe gesture ona table or couch surface and insantiates interactors for controlling devices in the living room" [23].

III. The Problem

A. Bluetooth Background

Mobile being the integral part of human life in recent times, it has also gained importance with the communication purposes it is used with. One such important local wireless communication is the Bluetooth communication [12]. Bluetooth came into existence with an idea of replacing cable protocol for wireless connectivity and later extended its applications to voice/data access points and personal ad hoc networks. Bluetooth being used for transferring data and other communication purposes in a certain range only and with no third party channel was essentially believed to be a trusted and reliable mode of communication. Later, with the increasing use of Bluetooth for various purposes and in different environments, vulnerabilities attacked Bluetooth with a wide variety of security threats. A Simple Pairing Protocol presents an interesting challenge to formal verification methods because it relies on out-of-band, human authentication [12]. This Simple Pairing Protocol uses plain, unauthenticated Diffie-Hellman key exchange. Authentication relies on an interesting out-of-band mechanism. Each device computes a short cryptographic hash of the established key and displays it on the device's screen. The two devices' owner(s) visually compares the displayed values and manually confirms that they match. In other words, authentication is done via key confirmation on a secure human channel, i.e., a human "equality oracle".

A simple flowchart of Bluetooth functionality is shown below in the figure:

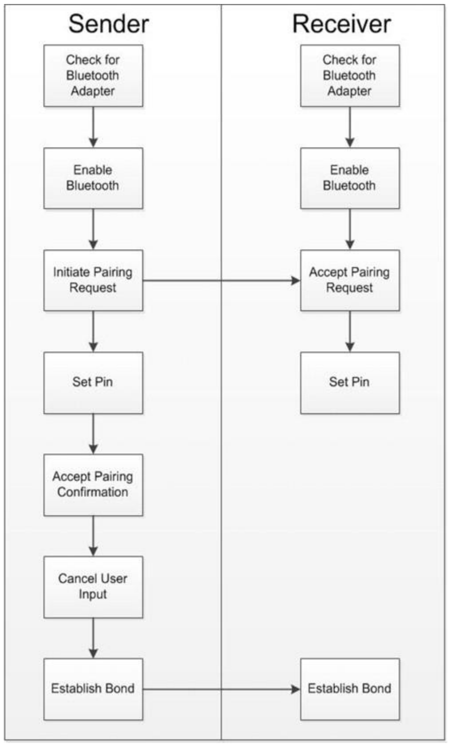

Figure 8, Flowchart describing Bluetooth connection functionality [21].

The authentication part of the Bluetooth with 'Simple Pairing Protocol' is complemented with the figure above depicting the generic and technical aspect of user authentication for pairing two devices for Bluetooth.

B. Bluetooth Command and Control Channel

A command and control (C&C) channel is evidence of the vulnerability exploits on Bluetooth technology. This is a kind of physical channel that is established between two devices that are in certain range and is responsible for data transfer between them. The Bluetooth C&C channel doesn't require any authorization and authentication to get connected to a device in the range. This is kind of useful for faster communication and data transfer when in a range without any user involvement but is mystifying the concept of security in the once most reliable communication technology. This C&C channel is mostly used in the mobile botnets (which can provide hackers with root permissions over a compromised device).

This Bluetooth C&C connection is similar to the regular Bluetooth connection on two devices, differing only by the pin generation on the devices asking for communication which involves human interaction, though is generally overlooked—making it vulnerable to attacks.

Considering the example of the Bluetooth C&C Channel as a vulnerability, a threat analysis could be derived. The user of the Bluetooth service on the device who wants to connect to another device could be a normal user or a hacker. With Bluetooth being one of the major device-to-device communication methods used in recent times in this mobile era, the vulnerability discussed above poses a serious threat. The effects of this threat are information theft, which may also include sensitive information and misuse of gathered information for various purposes. Hence the threat is to be considered seriously and a solution should be proposed once the threat is analyzed.

From the perspective of the five security objectives (confidentiality, integrity, authenticity, availability, and non-repudiation), the threat's causes and effects in the case of Bluetooth connection in automobiles will be discussed. Due to the minimal and easy authentication processes on a device, initializing connection through Bluetooth availability is guaranteed, though the integrity of the information can be compromised. When a smartphone and vehicle are connected via Bluetooth, the vehicle has free and open access to the data on the phone (following the initial pairing). By extension, anyone with access to this Bluetooth connection will have access to this data. The data is still accessed through the Bluetooth service rather than the mobile phone itself, however, possibly resulting in a denial of service issue.

Further concerns with Bluetooth connectivity can be attested to false authentication (as can be present in the automotive domain, as a pairing code is only required at the initial connection of device to vehicle). If a malicious attacker gains access to one's data via a Bluetooth connection, the user's device data and information captured in storage are almost fully vulnerable to the attacker (unless somehow encrypted or protected through methods not known by the average user). This presents a reasonable threat in today's society, as many people keep personal and confidential information on their mobile devices.

In addition to the general threat of losing one's personal information, an attacker may be able to send and receive audio between themselves and a Bluetooth-enabled car stereo—a further invasion of the user's privacy, outside the reach of their stored data [24]. Attackers may also be able to remotely access a user's phone and use its features, "including listening to calls, forwarding incoming calls, placing calls and sending text messages—and the user doesn't realize what's happening" [24]. With this many prevailing threats to Bluetooth connection security, a safer way of connecting one's device to their car stereo must be implemented.

IV. The Solution: Grip Gesture and Bluetooth Authentication in Automotive Context

A. Pressure Sensors in the Automotive Domain

The theoretical inflection for pressure sensors in automotive contexts is not a new idea. The rise of semi-autonomous vehicles has seen renewed interest in driver attentiveness [26]. Tesla Motors has placed a safeguard with the steering wheel to ensure that even with the AutoPilot feature enabled, users continue to keep their hands on the wheel [17]. An alternative to Tesla's process, patents have already been granted in the United States for pressure sensors in the steering wheel [27]. Europe has also been considering pressure sensors in motor vehicles as well.

German engineering firm Hoffman and Krippner, in collaboration with Guttersberg Consulting have proposed a method to detect driver drowsiness or medical emergencies with pressure sensors in the steering wheel [7].

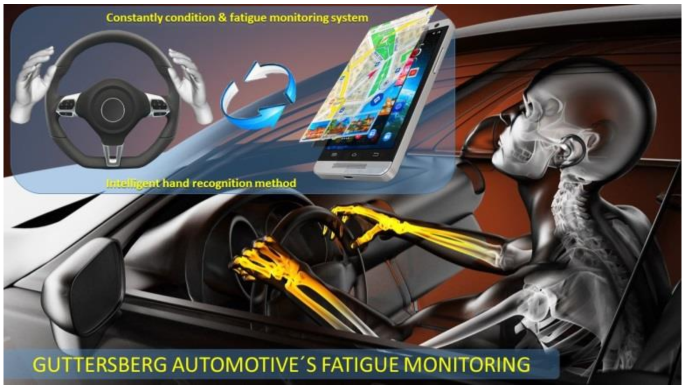
Constantly condition & fatigue monitoring system
Intelligent hand recognition method
GUTTERSBERG AUTOMOTIVE´S FATIGUE MONITORING

Figure 9, Showing intended functionality of patented pressure sensors

As the user becomes incapacitated, the grip on the steering wheel slackens and the vehicle responds accordingly. It is in this way that we hope our own proposal takes into account pragmatic considerations. If pressure sensors already exist within a vehicle, they can be adapted to also provide mobile user authentication as well.

B. A Typical Phone Pairing Framework with Our Approach

Initial Pairing:

1. The user enters a new vehicle and executes a pairing command from some physical button press in the vehicle.

2. The vehicle prompts the user to turn on Bluetooth on their smartphone and login to their smartphone using their personal password.

3. The vehicle prompts the user to grip the steering wheel at 10-2, 9-3, and 8-4 and builds a unique grip profile of the user. The unique grip profile of the user is processed by the vehicle and stored on the user’s phone. Because the user is required to log in to their phone every time they

wish to make a new grip profile, we ultimately achieve multi-factor authentication without the time requirements of more traditional multifactor methodologies.

Use Following Initial Pairing:

1. The user enters the vehicle with the Bluetooth on their phone activated. On power up, the vehicle senses the Bluetooth connectivity and enters a wait state until the user places their hands on the steering wheel of the vehicle.

2. Pressure sensors in the steering wheel detect the user's grip and perform a handshake routine with the user's phone to determine if the grip is authentic

   a) If the grip is determined to be authentic, the Bluetooth command and control channel is allowed to be formed.

   b) If the grip is not authentic, the car will alert the driver that pairing is not possible with the current grip profile.

3. In the background, the vehicle will poll the user's phone to recheck the grip profile every five minutes. If a failure threshold for number of failed verification attempts is reached, the vehiclewill prompt the user to place their hands on the wheel so that they can be re-verified. Failure to re-verify will result in the termination of the vehicle/smartphone pairing

Following Period of Inactivity:

1. If the user does not use the vehicle for a period of one month, the vehicle will execute an erase protocol that erases the unique smartphone identifier to that particular smart phone and the user will have to create a new profile when he/she enters the vehicle again

An architectural outline and control flow graph displays how phone pairing would occur using our system:

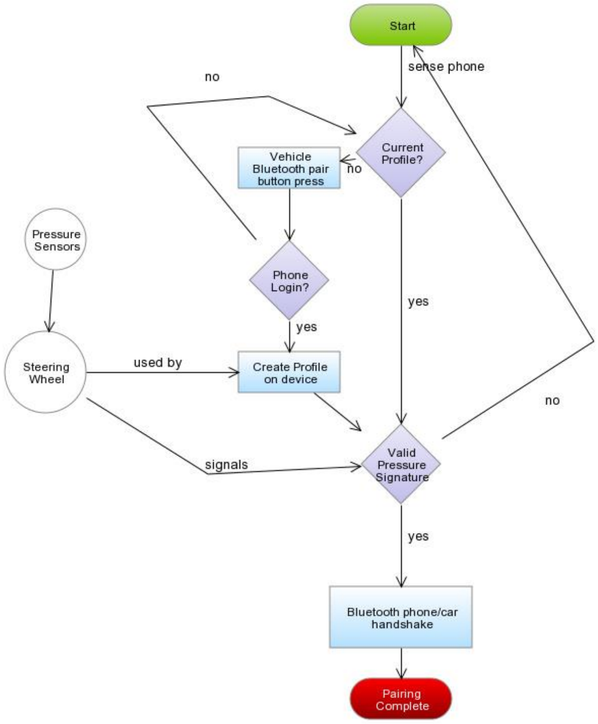

Figure 10, Showing the control flow and basic architecture of phone vehicle pairing system.

The visual model of our architecture (Fig. 10) betrays some challenges for the development team in their implementation of our design:

- Drivers being generally accustomed to holding the steering wheel in a particular fashion might damage the pressure sensors due to the frequent gripping on that specific region of the steering wheel. Placing the pressure sensors on the steering wheel in a skilled way so that frequency of gripping on a particular area would not damage the pressure sensors is an identifiable challenge for the designers.

- The “polling” process designed for the verification purpose could be challenging for the developers as there should be a proper synchronization maintained between the vehicle and the smartphone. Any delay or malfunction could terminate the connection and request for establishing new connectivity which could lead to frustration of the user—particularly if the user currently engaged in a phone call.

- In the scenario where there is a family creating multiple profiles, managing between different profiles could prove to be difficult in terms of design and development.
- Security of the user profile pressure signatures on the device would need to be accounted for. Not only is the Bluetooth connection capable of being intercepted, but the device itself may be vulnerable to the kinds of attacks found in the literature survey. Industry encryption methodologies like hash value storing should be incorporated into the design. This could cause challenges for the developers of the system.
- The mobile device ready to pair using the pressure sensors should be compatible to interact with the proposed interface of the vehicle.

V. Conclusion

In this paper, a novel process for securing the command and control channels in Bluetooth and automotive phone pairings is presented. Although the theoretical groundwork has been laid, additional research must be conducted to analyze the cost feasibility of adding pressure sensors to automobiles. Additionally, the sensitivity of pressure sensors has not seen much work in the literature. It is still uncertain if such hardware will be able to build a unique and reliable profile of the user in a steering wheel form factor. What is certain through the work in this paper is the need to secure the command and control channel between smartphones and automobile Bluetooth after initial pairings. Information leakage in automotive contexts continues to be a problem. Work has already shown that smartphone information - your phone number, call logs, recent contacts - can be stolen from rental cars if users fail to erase this data prior to returning the vehicle [11]. Questions arise as to what can happen if Bluetooth pairing was spoofed by attackers as well.

REFERENCES

[1] M. E. Ali, T. Hashem, A. Anwar, L. Kulik, I. Ahmed, and E. Tanin, “Protecting mobile users from visual privacy attacks,” *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing Adjunct Publication - UbiComp '14 Adjunct*, pp. 1–4, 2014.

[2] P. Andriotis et al., "A study on usability and security features of the Android pattern lock screen", Information & Computer Security, Vol. 24 Iss 1 pp. 53 - 72, 2016.

[3] A. Aviv et al., "Smudge Attacks on Smartphone Touch Screens," in *USENIX Workshop on Offensive Technologies*, 2010.

[4] T. Bradley, “Smartphone Security Thwarted by Fingerprint Smudges,” *PCWorld*, 11-Aug-2010.

[5] H. Chiang and S. Chiasson, “Improving user authentication on mobile devices: a touchscreen graphical password,” in *Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services (MobileHCI '13),* ACM, 2013, pp. 251-260.

[6] L. Constantin, “Most Android phones can be hacked with a simple MMS message or multimedia file,” *PCWorld*, Jul 27, 2015.

[7] B. Coxworth, “Smart steering wheel detects driver drowsiness,” *Gizmodo*, 10-Jul-2015.

[8] “Guttersburg Automotive,” *Guttersburg Automotive*. [Online]. Available at: http://guttersberg-automotive.com/. [Accessed: 16-Apr-2016].

[9] How to Control Bezel-Free Phones? ZTE Corporation, 2015.

[10] S.-H. Kim, J.-W. Kim, S.-Y. Kim, and H.-G. Cho, “A new shoulder-surfing resistant password for mobile environments,” *Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication - ICUIMC '11*, 2011.

[11] K. Komando, “One huge mistake people make when renting cars,” *USA Today*, 03-Jul-2015. [Online]. Available at: http://www.usatoday.com/story/tech/columnist/komando/2015/07/03/komando-car-rental-mistakes/29614165/.

[12] B. K. Mandal, D. Bhattacharyya, and T.-H. Kim, “An Architecture Design for Wireless Authentication Security in Bluetooth Network,” IJSIA International Journal of Security and Its Applications, vol. 8, no. 3, pp. 1–8, 2014.

[13] F. Manjoo, “A Murky Road Ahead for Android, Despite Market Dominance,” *New York Times*, 27-May-2015.

[14] M. P. Mattson, “Superior pattern processing is the essence of the evolved human brain,” *Frontiers in Neuroscience,* vol. 8, no. 265, Aug. 2014.

[15] W. Meng, W. H. Lee, and S. Krishnan, "A Framework for Large-Scale Collection of Information from Smartphone Users based on Juice Filming Attacks," in *Singapore Cyber-Security Conference (SG-CRC),* iOS

[16] D. H. Mortensen, “Eyes, grip and gesture as objective indicators of intentions and attention,” Proceedings of the 12th ACM international conference adjunct papers on Ubiquitous computing - Ubicomp '10, 2010. Press, 2016, pp. 99–106.

[17] “Model S Autopilot Press Kit,” *Tesla Motors*. [Online]. Available at: https://www.teslamotors.com/presskit/autopilot. [Accessed: 15-Apr-2016].

[18] K. Murao, “Mobile Phone User Authentication with Grip Gestures using Pressure Sensors,” in *Proceedings of the 12th International Conference on Advances in Mobile Computing and Multimedia*, New York, NY, 2014.

[19] H. Pieterse and M. Olivier, “Bluetooth Command and Control Channel,” *Computers & Security 45* (2014), p. 75-83, June 6 2014.

[20] D. Reisinger, “Android Security A Glaring Problem: 10 Reasons Why,” *Eweek (2014*), p. 1, Mar 25, 2014.

[21] M. Ryan, “Bluetooth: With Low Energy Comes Low Security”, *Presented as part of the 7th USENIX Workshop on Offensive Technologies*, Washington DC, Aug. 2013.

[22] A. Shah, P. Ved, A. Deora, A. Jaiswal, and M. D'silva, “Shoulder-surfing Resistant Graphical Password System,” *Procedia Computer Science*, vol. 45, pp. 477–484, 2015.

[23] T. Smirnova, “Grippo: Using Grip Gestures to Repurpose Everyday Objects as Controllers,” M.S. thesis, Comp Sci, Dept., RWTH Aachen University, 2015.

[24] A. Stern, “Bluetooth Connectivity Threatens Your Security,” Kaspersky Blog, Apr 15 2013. [Online]. Available at: https://blog.kaspersky.com/bluetooth-security/1637/.

[25] H. Sun, K. Wang, X. Li, N. Qin, and Z. Chen, “PassApp,” *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services - MobileHCI '15*, pp. 306–315, 2015.

[26] D. Szondy, “2016 Nissan Maxima keeps an eye on drowsy drivers,” 04-Apr-2015.

[27] Tk Holdings, Inc., "Steering Wheel with Hand Pressure Sensing", US8983732B2, 2015.

[28] S. Uellenbeck et al., "Quantifying the Security of Graphical Passwords: The Case of Android Unlock Patterns," in *Proceedings of the 2013 ACM SIGSAC Conference of Computer & Communications Security*, New York, NY, 2013.

[29] L. Whitney, “Android market share stays steady in US but sinks deeper in Europe,” *Cnet*, 02-Sep-2015. [Online]. Available at: http://www.cnet.com/news/android-market-share-stays-steady-in-us-but-sinks-deeper-in-europe/.

[30] S. Wiedenbeck, J. Waters, L. Sobrado, and J.-C. Birget. “Design and evaluation of a shoulder-surfing resistant graphical password scheme,” in Proc. of AVI 2006, pages 177–184, 2006.

[31] L. Wu et al., “The Impact of Vendor Customizations on Android Security” in *ACM Conference on Computer and Communications Security*, Berlin, Germany, 2013.