VolumePatterns: Using Hardware Buttons beyond Volume Control on Mobile Devices

Yasmeen Abdrabou
Bundeswehr University Munich
yasmeen.essam@unibw.de

Sarah Prange
Bundeswehr University Munich
sarah.prange@unibw.de

Lukas Mecke
Bundeswehr University Munich
lukas.mecke@unibw.de

Ken Pfeuffer
Bundeswehr University Munich
ken.pfeuffer@unibw.de

Florian Alt
Bundeswehr University Munich
florian.alt@unibw.de

CHI 2020 Workshop on Authentication Beyond Desktops and Smartphones.
CHI '20, April 25–30, 2020, Honolulu, HI, USA.

Abstract
While hardware buttons on mobile devices are mainly used for simple shortcuts, they offer more potential for quick, subtle and eyes-free input. In this work, we present *VolumePatterns* as an example prototype where users authenticate on their phones with short and long press volume button patterns.

Author Keywords
Hardware Buttons; Mobile Devices; Micro Authentication Tasks

CCS Concepts
•**Human-centered computing** → **Human computer interaction (HCI);** *Haptic devices;* User studies;

Introduction
For keyboards, controllers and mobile devices, hardware buttons are one of the most common input methods found on digital devices as their physicality affords intuitive and tactile interaction [5, 6]. On the smartphone, most of the user's interaction needs are handled via the touchscreen providing both, input and output capabilities. Physical buttons have remained to easen access to system features, but are underexplored since the inauguration of touch inputs.

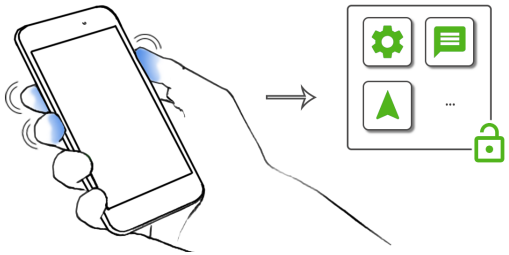

Figure 1: Hardware buttons can be used for a variety of purposes. In our work, we propose to use volume buttons for micro authentication tasks (i.e., allow for short tasks involving semi-private functionality) by entering patterns of volume button presses to unlock functionality like settings, messaging and navigation.

This research aims at revisiting mobile button interactions based on the current hardware design of smartphones. Most devices provide additional hardware buttons for simple input (e.g., power, volume control). Hardware buttons are beneficial in several ways: 1) they are eyes-free, thus allowing users to input without paying specific attention to it, 2) they allow for subtle input (i.e., users could in principle hit buttons, e.g., from inside a pocket), and 3) they can support microinteractions [8].

In this paper, we present *VolumePatterns* as an example use case of authentication through volume button patterns. As input on mobile touchscreen devices, be it for security sensitive issues or even casual interaction, is exposed to several risks and threat models (e.g., shoulder surfing [2] or smudge [1] attacks), we introduce this as a possible alternative with potential advantages over touchscreen input due to the ability of eyes-free and subtle interaction.

Volume Patterns

We present *VolumePatterns*, where users authenticate by doing a combination of volume button presses. We exclude the power button as in this context it is already assigned to lighting up the screen to trigger functionality worth protecting. In our prototype, *VolumePatterns*, we focus on unlocking as the main application in order to evaluate if this approach can provide additional security and would thus be suitable to unlock semi-private functionality.

Our idea of using volume keys to unlock a smartphone builds on previous work. The app *Volume Unlock* [7] allows to wake the screen with the device's volume buttons in case the power button is broken. With the app *Sequence Unlock* [4], it is possible to unlock the device with a sequence of volume button presses. However, this app requires root access and is not available over common software distribution platforms. It also has a very limited key space that we try to address in our approach.

We hypothesise our approach to be fairly resistant against shoulder-surfing due to the use of subtle finger movements. In addition, hardware button input does not leave traces on the display and is thus resistant against smudge attacks [1]. Furthermore, we benefit from the use of hardware buttons since users can interact with them eyes-free, thus enabling subtle and even out of sight authentication (e.g., in a pocket).

Theoretical Keyspace and Consequences

Due to the use of only two buttons (volume up and down), the theoretical keyspace for patterns is very limited. To increase the theoretical keyspace, we distinguish between long and short keypresses, hence there are four possible inputs:

(1) Volume Down ↓
(2) Volume Up ↑
(3) Long Volume Down ⇓
(4) Long Volume Up ⇑

Referring to classical pins of length four, this results in a theoretical key space of 256. Hence, the theoretical key space is too small to provide reasonable security for fully unlocking a device (especially in contrast to pins). However, the sequence length does not have to be limited to four input signs, which can increase the patterns' complexity.

An alternative would be to increase the input space further, e.g. by adding additional temporal distinctions (e.g. short, medium, long), allow for combinations of volume buttons (⇳, ↕) or include the power button (⏻) as input. For this work, we decide to keep the input simple to avoid mentally overloading the user, but kept the option of longer patterns.

Implementation
We implemented the concept as an Android app, mimicking a lock screen. Users can set a custom unlock pattern consisting of long and short volume key presses (i.e., from the input alphabet: ↓, ⇓, ↑, ⇑) with a minimal length of four. In case the user enters three wrong sequences, the app switches to a backup lock screen, where the user can unlock the phone with a custom numeric PIN (compare Figure 2).

To find a good threshold for the distinction between long and short key presses, we conducted a pilot test (N=20) where participants had to enter the pattern "⇑ ↑ ⇑ ↑", i.e. alternating long and short presses of the volume up button. Our results (compare Figure 3) show that all users in-

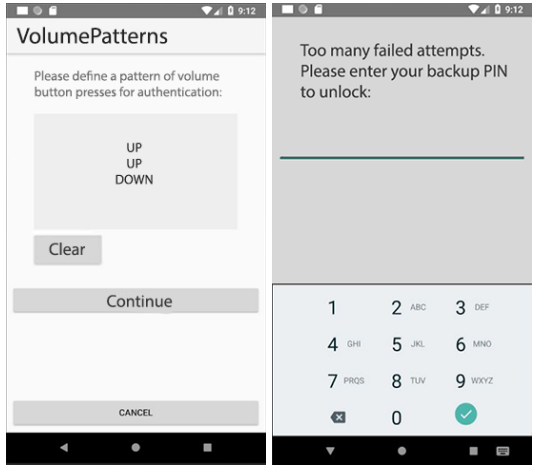

Figure 2: Sample screens of our prototype allowing the user to define a new pattern of volume key presses (left) and the fallback PIN screen in case of three failed attempts.

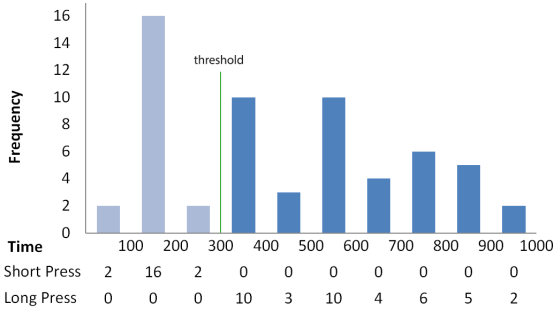

Figure 3: Distribution of long and short key presses. The results show, that 300ms is a valid threshold, as none of the short presses was longer (max=208ms) and none of the long presses shorter (min=328ms).

terpreted a “long press” as at least 300ms (mn=381.2ms, std=163.6), while the longest recorded “short presses” had a duration of 208ms (mn=144.4ms, std=24.8). From that we recommend a threshold of 300ms, which is also aligned with the long press duration of the Google keyboard (Gboard) [3] that is pre-installed on many Android devices.

Conclusion and Future Work

In this work we proposed a prototype called *VolumePatterns* to authenticate on a mobile device using patterns of volume button presses. We did a pilot test (N=20) to identify a threshold for long and short presses differentiation which tended to be 300ms as the long press duration of the Google keyboard (Gboard). We reported the concept and implementation of our *VolumePatterns*. As a concrete next step, we will evaluate the prototype to test its usability and security against attacks (i.e. shoulder surfing).

REFERENCES

[1] Adam J. Aviv, Katherine Gibson, Evan Mossop, Matt Blaze, , and Jonathan M. Smith. 2010. Smudge Attacks on Smartphone Touch Screens. In *WOOT'10 Proceedings of the 4th USENIX conference on Offensive technologies*. USENIX Association, Article No. 1–7.

[2] Malin Eiband, Mohamed Khamis, Emanuel von Zezschwitz, Heinrich Hussmann, and Florian Alt. 2017. Understanding Shoulder Surfing in the Wild: Stories from Users and Observers. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 4254–4265.

[3] Tech Entice. 2018. How To Increase The Long Press Delay In Google Keyboard for Android? **https://www.techentice.com/increase-long-press-delay-google-keyboard-android/**. (2018).

[4] Neil Gonzales. 2015. Unlock Your Android with a Secret Sequence of Volume Key Presses. (2015). **https://tinyurl.com/ybmoj4ob**

[5] Eve Hoggan, Stephen A. Brewster, and Jody Johnston. 2008. Investigating the Effectiveness of Tactile Feedback for Mobile Touchscreens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 1573–1582. DOI: **http://dx.doi.org/10.1145/1357054.1357300**

[6] Jeong Ho Kim, Lovenoor Aulck, Michael C. Bartha, Christy A. Harper, and Peter W. Johnson. 2014. Differences in typing forces, muscle activity, comfort, and typing performance among virtual, notebook, and desktop keyboards. *Applied Ergonomics* 45, 6 (2014), 1406 – 1413. DOI:**http://dx.doi.org/https://doi.org/10.1016/j.apergo.2014.04.001**

[7] TrishTech.com. 2017. Volume UnLock: Use Volume Buttons to Unlock Android Device. (2017). **https://tinyurl.com/y8wpdnvf**

[8] Katrin Wolf, Anja Naumann, Michael Rohs, and Jörg Müller. 2011. A Taxonomy of Microinteractions: Defining Microgestures Based on Ergonomic and Scenario-Dependent Requirements. In *Human-Computer Interaction – INTERACT 2011*, Pedro Campos, Nicholas Graham, Joaquim Jorge, Nuno Nunes, Philippe Palanque, and Marco Winckler (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 559–575.