Lane Gramling
EECS 700
Reading 20

Summary - PatternListener: Cracking Android Pattern Lock Using Acoustic Signals

This paper focuses on exploiting the behavior of reflected acoustic signals as a side-channel for guessing the pattern lock on Android phones. The attack gathers its information by playing an imperceptibly high-pitched acoustic signal at approx. 18kHz from the phone's speaker, while simultaneously using the microphone to detect perturbations in the acoustic signal as it is reflected off of the user's fingertip. Afterwards, the data is easily sent to a remote server where the recorded audio is pre-processed and computed on for candidate lock patterns. In their studies on 130 unique lock patterns, the authors found over 90% accuracy in guessing the pattern within five attempts.

Contributions

First, the proposal of utilizing acoustic signals for observing finger movement during unlocking can be regarded as a novel contribution, as the authors point out that all past works they found have only explored using wi-fi signals, motion sensors, and other side-channels to observe movement around the phone. This is particularly useful in this application because the acoustic signals used attenuate strongly with distance and time, eliminating much potential for noise and interference caused by the surrounding spatial environment of the phone beyond close range.

Another major contribution of the paper is the proposed Turning Points Identification (TPI) algorithm, which reconstructs candidate patterns from the pre-processed audio data approximating finger movements and is at the heart of PatternListener's capabilities. This algorithm takes the cophase and orthogonal audio signals which have been downsampled and run through a low-pass filter for pre-processing, and uses the Local Extreme Value Detection (LEVD) signal processing algorithm to pinpoint turning points in the finger's movement. After this, the startpoints and endpoints of the movement sequence are identified, which provides enough information to segment the individual movements. Pattern lines are inferred by extracting a number of features from the phase-shift information in the segments to deduce movement distance and direction, finally constructing a pattern tree from the feature vectors to narrow down probable candidate patterns.

Additionally, data is given on the impact of different variables corresponding to the environment, as well as a study on 130 unique patterns which showed success rates over 90% within five attempts. In this data, results are shown on tests in different physical environments with different noise levels, the effect of finger speed, and the effect of nearby objects both static and moving. It is observed that interference by moving objects becomes negligible after about 60cm, ambient noise levels do not have a significant impact on performance, and that more complex patterns yield better protection from the attack.

Limitations

Overall, this attack seems quite effective given the information provided in the paper. One noticeable limitation is that there seemed to be little to no testing done on varying hand/finger positions, which could feasibly show more limited results; likewise, positions where the hand may inadvertently cover the speaker or microphone could have a similar effect. Even still, given the amount of times a user is bound to unlock their phone in a given day, compounded with the widespread norm of how people hold their phones, it is highly likely with the shown success rates that it wouldn't be even a day before an ideal scenario for PatternListener arises.

Future Work

This paper expressed an attack that is largely independent of phone model and OS version, and exhibits high success rates under the tested scenarios. Additionally, due to the widespread popularity of microphone permissions and invisibility to the user, it is very easy to remotely deploy with success. As such, future work should be directed towards exploring countermeasures. One possible suggestion would be for the Android OS or component manufacturers to limit the speakers from playing frequencies beyond the audible range of humans, or force some type of visual notification that the speaker or microphone is

on. The philosophy of this is similar to the now common practice of having an LED light up next to the lens when a phone's camera is in use, or how it is mandated in China (if I recall correctly) that a shutter sound is forced upon taking a picture.