Breaking the Android Pattern Lock Screen with Neural Networks and Smudge Attacks

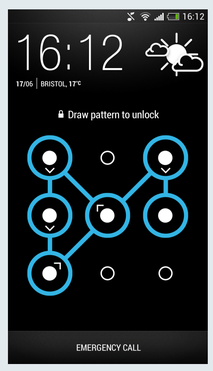
Panagiotis Andriotis, Theo Tryfonas, Zhaoqian Yu

Motivation

The Android pattern lock screen is a popular graphical authentication scheme. Apps made for other O.S. mimic the functionality and usability it offers and the method is spreading to other areas of interest to cover various needs (e.g. user authentication for mobile banking apps).

Existing methods to bypass the security mechanism require the USB Debugging Mode to be enabled.

Other methods need the use of the ADB tool to be installed from the Android SDK (available at the official Developers' portal) and also root privileges to remove files like "*.key" from specific folders existing in the internal structure of the device (e.g. in data/system).

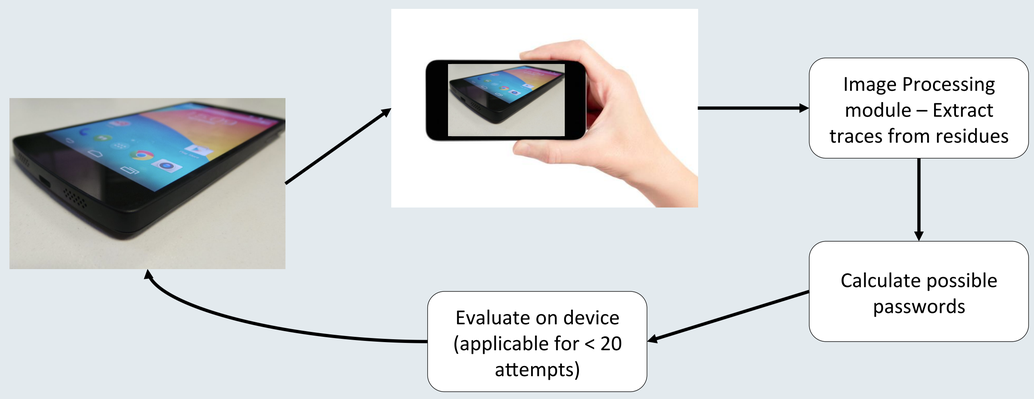


Known Vulnerabilities:

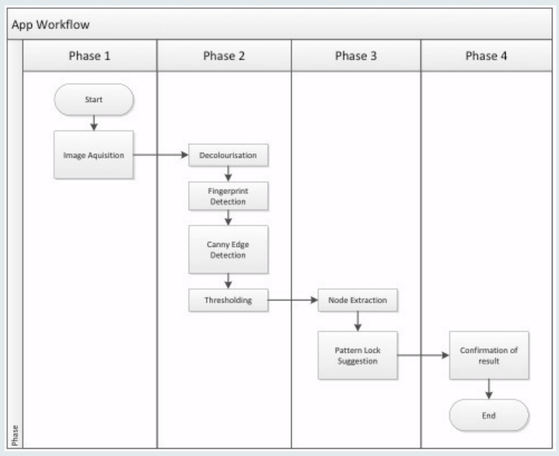
- Shoulder surfing
- Smudge attacks

Can we use prior knowledge on heuristic rules that define pattern formation with smudge attacks to bypass the security scheme without the use of 'adb' or 'su'?

Methodology



We are developing an app (which will be the final deliverable) able to perform the following work circle: a) Take a hi-res picture of the screen under certain circumstances [3], b) Process image using various algorithms (described below), c) Calculate a list with possible patterns based on an ANN approach and on information gathered from b (extracted nodes, directionality). The evaluation part is not automated. The investigator follows the recommendation list until the device becomes accessible. (There is an OS limitation that totally locks the phone after 20 non successful inputs.)

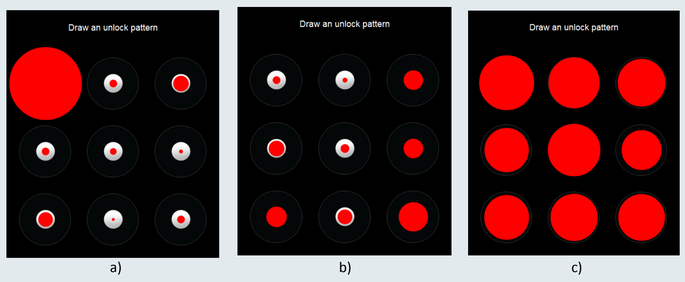


- In general we aim to extend existing OpenCV functions.
- Decolourisation: Choose upon different implementations of OpenCV on greyscaling.
- Fingerprint detection (as part of a feature detection process) to estimate directionality.
- *Canny* Edge Detection in order to extract the contour of the trace.
- Thresholding to separate objects from background using the *Otsu* algorithm.

Survey data reveal biased input

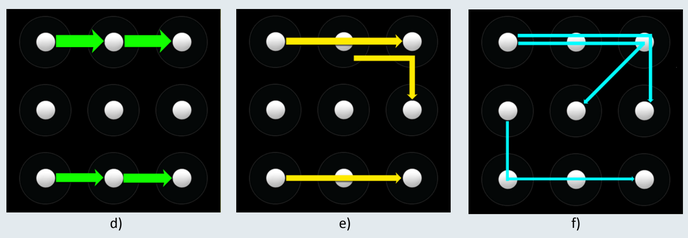
WiSec13 – Web Survey [1]

a) Start point frequency
b) End point frequency



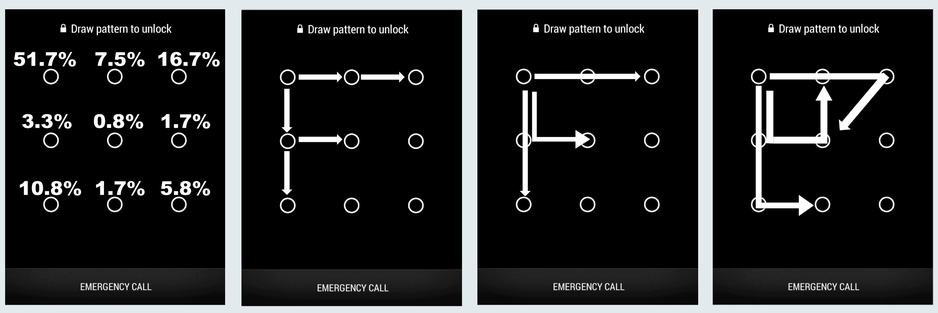
c) Monogram appearance

d) Popular Bigrams
e) Popular Trigrams



f) Popular Fourgrams

HCII2014 – Android App [2]



Different samples of Android users provided the same answers in different periods of time, showing that there exist human driven rules that dictate biased input during the pattern lock screen formation. Exploitation of these rules in collaboration with the outcome of automated smudge attacks (using neural networks on our existing pattern dataset) will produce a list of possible passwords. The goal is to minimize the possibility the phone to be locked after a series of invalid inputs and provide a forensically sound solution to access the device without gaining super-user privileges.

Pattern lock screen Security Evaluation

Although the overall password space of the Android pattern lock screen consists of 389112 different passwords, the results of our studies show that if we take into account biased input, it is possible to limit possible passwords and achieve our goal.

Below: The password space.
Right: possible patterns if we know one or two nodes from the pattern.

Length	Unique Patterns
4	1624
5	7152
6	26016
7	72912
8	140704
9	140704

Attributes	Length	Unique Patterns
Starts at Node 0	4	154
	5	684
	6	2516
	7	7104
	8	13792
	9	13792
Starts at Node 0 and Crosses Node 4	4	82
	5	456
	6	1948
	7	6152
	8	12944
	9	13792

Reduced password space using more traces features (e.g. knight moves and trigrams).

Length	Unique Patterns No Knight Moves	Unique Patterns No Knight Moves + (012)
4	44	1
5	160	6
6	442	20

- Improvements could be made to the initial concept to include a hybrid approach when we design the neural network (NN) to include 'fuzzy' logic. Such a system will be able to incorporate a multilayer perceptron (MLP) NN with decision models containing expendable IF-THEN rules adjusted to the pattern lock screen case (e.g. IF a pattern consists of knight moves, THEN the likelihood is 40%.)
- Also, the image processing module can be further improved if we know the constrained environment where a pattern lives (depending on the device model).

References

[1] P. Andriotis, T. Tryfonas, G. Oikonomou, and C. Yildiz. A pilot study on the security of pattern screen-lock methods and soft side channel attacks. In *Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks*, pages 1–6. ACM, 2013.

[2] P. Andriotis, T. Tryfonas, and G. Oikonomou. Complexity Metrics and User Strength Perceptions of the Pattern-Lock Graphical Authentication Method. In Theo Tryfonas and Ioannis Askoxylakis (Eds.), *Human Aspects of Information Security, Privacy, and Trust*, pages 115-126, Crete, Greece, July 2014, Springer-Verlag. Lecture Notes in Computer Science 8533.

[3] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith. Smudge attacks on smartphone touch screens. In *Proceedings of the 4th USENIX conference on Offensive technologies*, pages 1–7. USENIX Association, 2010.

Acknowledgement

This work has been supported by the European Union's Prevention of and Fight against Crime Programme "Illegal Use of Internet" ISEC 2010 Action Grants, grant ref. HOME/2010/ISEC/AG/INT-002 and the Systems Centre of the University of Bristol.

Digital Forensics Tools

University of BRISTOL