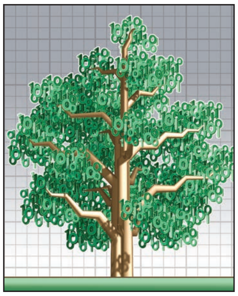
Editor: Munindar P. Singh • m.singh@ieee.org



# Is Bigger Better When It Comes to Android Graphical Pattern Unlock?

**Adam J. Aviv** • *United States Naval Academy*

**Ravi Kuber** • *University of Maryland, Baltimore County*

**Devon Budzitowski** • *United States Naval Academy*

Android unlock patterns are likely the most prevalent graphical password system to date. However, human-chosen authentication stimuli (such as text passwords and PINs) are easy to guess. Does increasing the grid size from 3 × 3 to 4 × 4 help the situation? Yes and no.

Researchers have proposed a range of graphical password mechanisms,[1] but none have become as prevalent as the Android's graphical pattern unlock. The unlock pattern, or password pattern, is perhaps the most widely used graphical password system to date. Coming preinstalled on all Android phones, it's the choice of a surprising number of Android users.[2]

The password pattern requires users to recall a "pattern" drawn by connecting a set of 3 × 3 contact points arranged in a square grid. A stroke-based pattern must be drawn such that the pattern can be completed without lifting and connect at least 4 contact points, without avoiding any intermediate points. Despite there being 389,112 possible patterns, users select patterns from a much smaller set that are easily guessable, roughly at the same rate as a random 3-digit PIN.[3]

One intuitive and straightforward method for increasing the entropy of user patterns is to increase the number of contact points: Why not allow a 4 × 4, 5 × 5, or even larger grid for users to select patterns? For example, even expanding the grid to 4 × 4 increases the number of possible patterns to 4,350,069,823,024. It's reasonable to expect that expanded grids would also increase the complexity of user chosen patterns.

As part of a series of research papers investigating Android patterns,[4-7] we examined the research question: Does increasing the grid size increase the security of human-generated patterns? We conducted an extensive in-person and online study in support of the research, finding that in some cases, yes – expanded grid sizes can produce stronger patterns, but in many cases, users still choose from a predictable and guessable set. This supports ample anecdotal evidence from password research that humans are poor at selecting strong passwords.[8]

In this article, we review some of the major results of our research and provide insights into future directions.

## Data Collection

We conducted two studies to collect Android patterns on both 3 × 3 and 4 × 4 grid spaces. Realism was an important consideration when conducting the research. Although leaked datasets containing real-world, text-based passwords are publicly available, no such resources are available for Android patterns, nor should we expect them to become available. Android patterns are used

1089-7801/17/$33.00 © 2017 IEEE

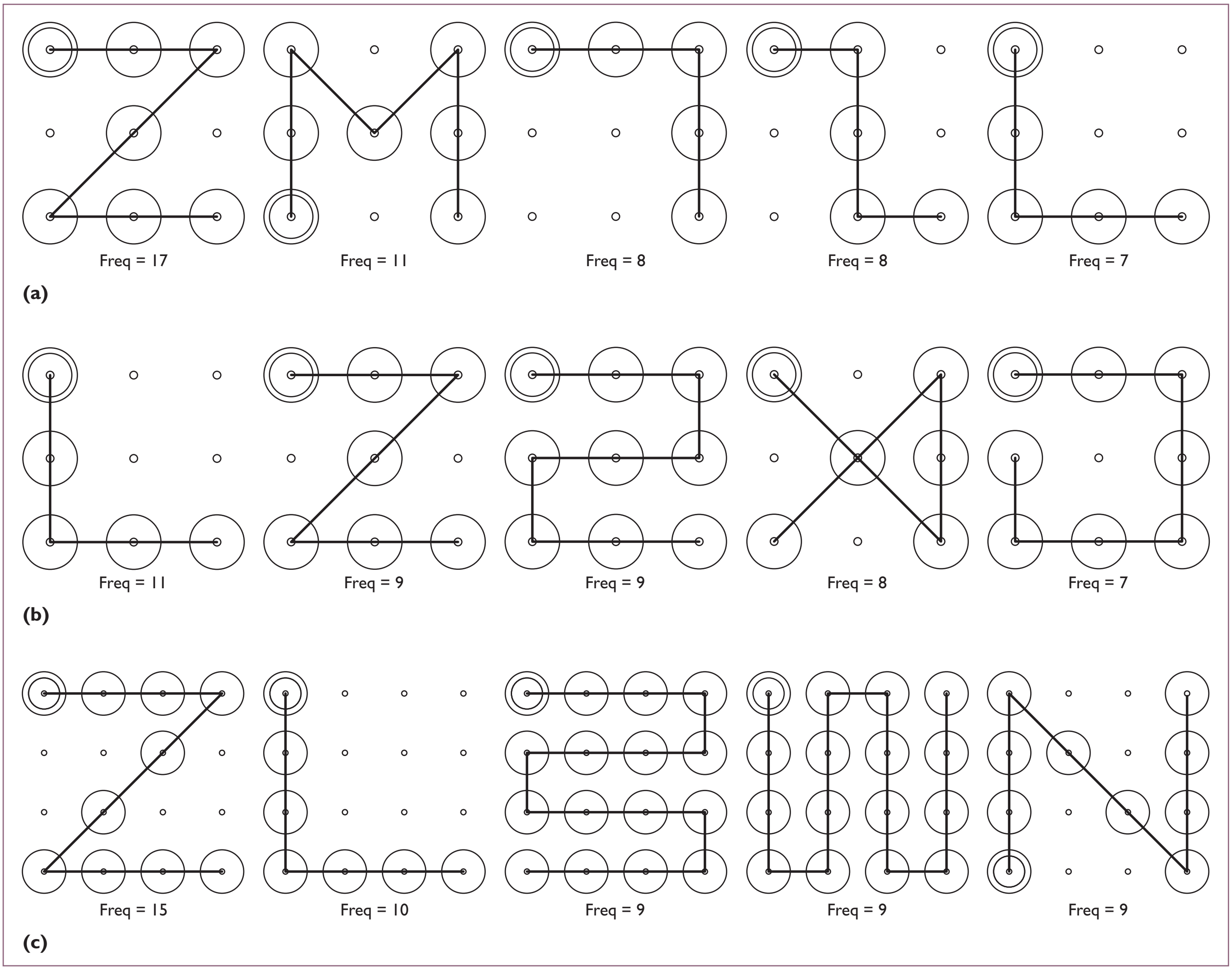


*Figure 1. Top 5 most frequently occurring patterns. (a) Self-reported 3 × 3. (b) Pen-and-paper (pen-paper) 3 × 3. (c) Pen-paper 4 × 4.*

for local authentication, so there can't be large leaks — and further, the use of 4 × 4 grid patterns is uncommon among Android users. As such, we had to develop new strategies for collecting these patterns.

First, as password patterns are graphical in nature, we developed a pen-and-paper (pen-paper) protocol built upon related work[3] by which participants, literally, draw a set of patterns that they wish others "not to guess" (so-called *defensive patterns*) and then a set of patterns that they believe others choose (so-called *offensive patterns*). If the participant chooses either a pattern that others didn't guess or were able to guess a pattern that others chose, they received a reward.

The pen-paper study was conducted with 80 participants in 10 sets of focus group. These took place over a period of 6 weeks. The participants generated 494 3 × 3 patterns (380 offensive and 114 defensive) and 504 4 × 4 patterns (385 offensive and 119 defensive). Some patterns were rejected from the analysis because they didn't follow pattern-generation rules.

As these drawn patterns might not conform perfectly to real users patterns, we also conducted a large online study of self-reported patterns to compare against the in-person study. Using Amazon Mechanical Turk, we recruited 440 participants who we confidentially asked to report their Android pattern or features of their patterns.

We performed a number of comparisons between the datasets to ensure that self-reported and drawn patterns are similar,[6] finding overall similarity in the datasets in a number of ways, as we'll discuss.

## Data Characterization

As a first step in the analysis, we can see in Figure 1 that participants

*Table 1. Fraction of repetitions, symmetries, and the embedding of 3 × 3 patterns in 4 × 4 patterns.*

| Pattern | Size | Repetitions | Symmetries | Embedding |
|---|---|---|---|---|
| Self-reported 3 × 3 | 440 | 203 (46.1%) | 336 (76.36%) | N/A |
| Pen-paper 3 × 3 (all) | 491 | 245 (49.9%) | 398 (81.1%) | N/A |
| Pen-paper 3 × 3 (offensive) | 378 | 187 (48.3%) | 309 (79.8%) | N/A |
| Pen-paper 3 × 3 (defensive) | 113 | 16 (14%) | 54 (47%) | N/A |
| Pen-paper 4 × 4 (all) | 501 | 179 (35.7%) | 204 (40.7%) | 166 (33.1%) |
| Pen-paper 4 × 4 (offensive) | 382 | 156 (40.8%) | 177 (46.3%) | 142 (37.1%) |
| Pen-paper 4 × 4 (defensive) | 119 | 10 (8.4%) | 10 (8.4%) | 24 (20.1%) |

*Table 2. Statistics of the length measures (mean [$q_l$:$q_l$]).**

| Pattern | Length | Normalized length | Normalized stroke length |
|---|---|---|---|
| Self-reported 3 × 3 | 6.0 [5:7] | 0.7 [0.6:0.8] | 2.9 [2:3.5] |
| Pen-paper 3 × 3 (all) | 6.3 [5:7] | 0.7 [0.6:0.8] | 2.9 [2.2:3.7] |
| Pen-paper 3 × 3 (offensive) | 6.3 [5:8] | 0.7 [0.6:0.9] | 3.0 [2.2:3.8] |
| Pen-paper 3 × 3 (defensive) | 6.0 [5:7] | 0.7 [0.6:0.8] | 3.0 [2.4:3.5] |
| Pen-paper 4 × 4 (all) | 9.6 [7:12] | 0.6 [0.4:0.8] | 3.2 [2.3:3.8] |
| Pen-paper 4 × 4 (offensive) | 9.8 [7:12] | 0.6 [0.4:0.8] | 3.2 [2.3:3.8] |
| Pen-paper 4 × 4 (defensive) | 8.8 [6:11] | 0.6 [0.4:0.7] | 3.0 [2.0:3.7] |

* The normalized length was calculated by dividing by the total available points, and the normalized stroke length was calculated by mapping the 3 × 3 and 4 × 4 grid on a 1 × 1 Cartesian plane.

tend to choose common, normally shaped patterns. Patterns shaped like letters, such as Z, L, N, or M patterns, are quite common in both 3 × 3 and 4 × 4 data, especially as you consider rotations and transformations, by which a pattern can be rotated or flipped. For example, the Z- and N-shaped patterns are transformations of each other, a so-called *symmetry*.[4]

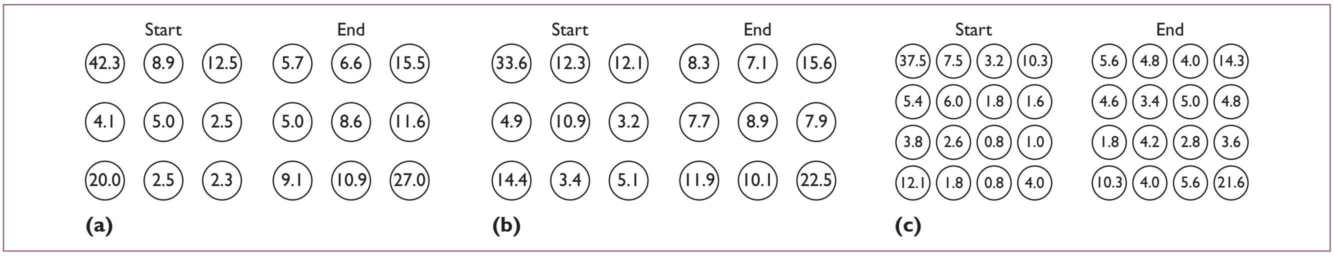
In fact, large portions of the dataset are symmetric, much more so than one would expect. As Table 1 shows, in some cases 50 percent of the data is symmetric to some other pattern within the dataset. In 4 × 4 patterns, a large fraction of the patterns is simply the embedding of 3 × 3 patterns, where a 3 × 3 pattern is mapped to the 4 × 4 grid space.

Further analysis of the pattern properties, such as length, reveal other similarities between the 3 × 3 and 4 × 4 pattern datasets. Table 2 displays the patterns' length properties, where *length* is the total number of contact points in the pattern. The normalized length is based on the number of contact points, and the normalized stroke length is the length of the lines in the pattern calculated by mapping the grid into a 1 × 1 Cartesian plane.

As you can see, both the distributions for the self-reported and pen-paper 3 × 3 patterns, and the length statistics (particularly the normalized ones) are similar to that of the pen-paper 4 × 4 pattern. This suggests, as apparent in the frequency data, that participants choose patterns of roughly the same properties in the 3 × 3 and 4 × 4 data.

Finally, we can measure the start and end conditions (which contact point a pattern starts with and which point it ends on). As Figure 2 shows, for both 3 × 3 and 4 × 4 patterns, there are strong tendencies for users to begin patterns in the upper-left of the grid, and end in the lower regions, particularly to the right. A random distribution of patterns would have equal likelihood for the start and end points.

All these properties of human-generated patterns — the repetitions, symmetries, and embedding — can all be leveraged by an attacker to inform a guessing strategy to attack



(a)

| Start | | | End | | |
|---|---|---|---|---|---|
| 42.3 | 8.9 | 12.5 | 5.7 | 6.6 | 15.5 |
| 4.1 | 5.0 | 2.5 | 5.0 | 8.6 | 11.6 |
| 20.0 | 2.5 | 2.3 | 9.1 | 10.9 | 27.0 |

(b)

| Start | | | End | | |
|---|---|---|---|---|---|
| 33.6 | 12.3 | 12.1 | 8.3 | 7.1 | 15.6 |
| 4.9 | 10.9 | 3.2 | 7.7 | 8.9 | 7.9 |
| 14.4 | 3.4 | 5.1 | 11.9 | 10.1 | 22.5 |

(c)

| Start | | | | End | | | |
|---|---|---|---|---|---|---|---|
| 37.5 | 7.5 | 3.2 | 10.3 | 5.6 | 4.8 | 4.0 | 14.3 |
| 5.4 | 6.0 | 1.8 | 1.6 | 4.6 | 3.4 | 5.0 | 4.8 |
| 3.8 | 2.6 | 0.8 | 1.0 | 1.8 | 4.2 | 2.8 | 3.6 |
| 12.1 | 1.8 | 0.8 | 4.0 | 10.3 | 4.0 | 5.6 | 21.6 |

*Figure 2. Frequency of pattern start and end points (in percent). (a) Self-reported 3 × 3. (b) Pen-paper 3 × 3. (c) Pen-paper 4 × 4.*

the security of graphical patterns, as we performed next to measure the security of patterns.

## Measuring Security of Patterns

In the literature of password research, the established method for measuring the security of passwords is based on *guessability*.[9] In its simplest representation, guessability metrics model an attacker's power, asking how many guesses would it take for an attacker, in an offline setting, to *guess* a password. The attacker's knowledge matters; for example, the attacker might have a model of how users select passwords that informs the guesses. The better the model (or the more non-random the passwords being guessed) the easier, or more guessable, the passwords.

We apply a similar notion to the data of graphical password patterns, known as *partial guessing entropy*.[10] Similar to guessability, partial guessing entropy attempts to measure how much randomness (or how hard it would be to guess) a fraction of a set of passwords. This is a common technique employed in prior work[3,11] for analyzing graphical passwords, and allows us to directly compare the relative strength of the different datasets we collected.

Crucial for this metric is developing a method for guessing patterns that accounts for how humans select patterns. Naively, we could simply guess patterns in some arbitrary order, from smallest to largest, based on numbering, and so forth, but this guessing strategy would likely perform poorly because it would take a substantially long time to enumerate the patterns that a human selected, focusing instead on coverage of more complex, less-likely-to-be-selected patterns. Based on a training set, we can instead design a guessing strategy that attempts to leverage the properties described earlier, namely that patterns repeat, are symmetric, and follow regular forms.

### Likelihood Estimators

In the development of a guessing strategy, we need to ensure that we don't over-train the data, so we tuned our algorithm using the pen-paper datasets, reserving the self-reported dataset as a *test set* to evaluate the performance of the routine on an independent set.

The guessing algorithm proceeds by assuming that there are some set of sample patterns to train on, and based on that training set, it generates an ordered set of guesses of patterns. We employed a Markov model likelihood estimator to do the training. Using a tri-gram model, we calculated the conditional transition probabilities that two tri-grams were connected in the pattern based on the examples in the training set. The model then calculates a comparable value representing the likelihood of a pattern, which speaks directly to how likely a user would be to choose that pattern.

For example, the Markov model indicates higher likelihood for patterns that start in the upper left and end in the lower right, as indicated by the training data. Further, it can recognize that patterns that include the top row of contact points, moving left to right, are more likely to be followed either by a diagonal set of contact points or a vertical set of contact points, rather than, say, connecting directly to the bottom middle contact point.

There isn't sufficient space to outline the specifics of the Markov model in this article. We refer the reader to our previous work[4] for details of the model. It suffices for this discussion to understand that the model, given a training set and a test pattern, produces a likelihood score. Further, given the model, we can also generate patterns that are likely to occur but might not be directly present in the dataset. These properties will be used in the guessing routine, described next.

### Guessing Algorithm

The algorithm uses the training data and the likelihood estimator, trained on that data, to guess a set of patterns. The goal of the guessing algorithm isn't to just guess as many patterns as possible, but as quickly as possible. The order of the guessing directly impacts the partial guessing entropy, which considers what fraction of the dataset is cracked after a certain number of attempts.

As described earlier, to develop the algorithm, we used the pen-paper

*Table 3. Partial guessing entropy comparisons.**

| Pattern | $\alpha = 0.1$ | $\alpha = 0.2$ | $\alpha = 0.5$ | Guessed total (%) | Guessed with 20 attempts (%) |
|---|---|---|---|---|---|
| **Self-reported 3 × 3** | **6.62** | **6.95** | **9.49** | **95.9** | **15.0** |
| **Pen-paper 3 × 3 (all)** | **6.59** | **6.99** | **8.93** | **97.2** | **16.7** |
| Pen-paper 3 × 3 (offensive) | 6.98 | 7.69 | 9.31 | 95.3 | 12.5 |
| Pen-paper 3 × 3 (defensive) | 9.43 | 9.79 | 10.98 | 90.2 | 4.0 |
| **Pen-paper 4 × 4 (all)** | **6.23** | **6.64** | **11.61** | **66.7** | **19.9** |
| Pen-paper 4 × 4 (offensive) | 6.46 | 7.57 | 10.40 | 67.7 | 16.7 |
| Pen-paper 4 × 4 (defensive) | 6.23 | 6.64 | 11.61 | 37.4 | 3.2 |
| Uellenbeck and colleagues' 3 × 3 (offensive)[3] | 7.56 | 7.74 | 8.19 | | |
| Uellenbeck and colleagues' 3 × 3 (defensive)[3] | 8.72 | 9.10 | 10.90 | | |
| Song and colleagues' 3 × 3 (with meter)[11] | 8.96 | 10.33 | 12.29 | | |
| Song and colleagues' 3 × 3 (without meter)[11] | 7.38 | 9.56 | 10.83 | | |
| Random 3 × 3 pattern ($U_{389,112}$) | 18.57 | 18.57 | 18.57 | | |
| Random 4 × 4 pattern ($U_{4,350,069,823,024}$) | 41.98 | 41.98 | 41.98 | | |
| Random 6-digit PIN ($U_{1,000,000}$) | 19.93 | 19.93 | 19.93 | | |
| Random 5-digit PIN ($U_{100,000}$) | 16.60 | 16.60 | 16.60 | | |
| Random 4-digit PIN ($U_{10,000}$) | 13.29 | 13.29 | 13.29 | | |
| Random 3-digit PIN ($U_{1,000}$) | 9.97 | 9.97 | 9.97 | | |
| Random 2-digit PIN ($U_{100}$) | 6.64 | 6.64 | 6.64 | | |
| Real users' 4-digit PINs[11,12] | 5.19 | 7.04 | 10.08 | | |

* Here, $\alpha$ refers to the fraction of the dataset being guessed, and results are reported in bits of entropy with comparisons to other related work.

data as the training set, and to test the effectiveness we applied a five-fold cross-validation method on a random selection of 500 patterns from each dataset. The cross validation proceeds by treating four of the five folds as training data, testing on the remaining fold. The results are the average across each result by which each of the folds is treated as the test set.

After much iteration, we developed a guessing routine that best fit our training data. The guessing routine is first informed by the observation that due to the high number of repetitions, the most optimal first step is to guess all unique patterns in the training set, ordered based on repetition frequency, with ties in frequency broken by the likelihood metric. Next, we compute and guess all unique (not previously guessed) symmetries of the training data, again ordered based on the likelihood estimator. Although a large portion of the patterns in the test set are likely to be cracked at this point, we still need to generate more patterns using the Markov model to make 50,000 guesses in total. This process was able to guess 97.2 percent of the human-generated 3 × 3 patterns within 50,000 guesses. Note that there are 389,112 possible patterns.

For 4 × 4 patterns, we can use the same routine, but we can take advantage of additional training information, namely the 3 × 3 patterns. Recall from Table 1 that a large fraction of the 4 × 4 patterns is simply an embedding of 3 × 3 patterns into the larger grid space. So, for 4 × 4 patterns, we included 3 × 3 embedding in the second stage when guessing all the training data's symmetries. After doing so, we are able to crack 66.7 percent of the human-generated 4 × 4 patterns within 50,000 attempts (note that there are 4,350,069,823,024 possible 4 × 4 patterns).

Finally, with the guessing algorithm fixed, we can use all the 3 × 3 pen-paper data as training and attempt to crack the self-reported 3 × 3 patterns — the reserved test set. Table 3 presents the complete results, where we also include the percentage of patterns guessed after 20 attempts, the lockout point on most smartphones.

The results in Table 3 use values in terms of bits of entropy, which we can loosely translate into how much randomness occurs in the dataset (as it relates to how difficult it is to guess that dataset). Note that $\alpha$ refers to the fraction of the dataset guessed, so for example, to guess 50 percent of the self-reported data has 9.49 bits of entropy, which is comparable to guessing a random 3-digit PIN. For 4 × 4 patterns, guessing 50 percent of the patterns has an entropy of 11.61, which is slightly more challenging, but it's less challenging than guessing a 4-digit PIN. Overall, from these results, although there are cases where guessing 4 × 4 patterns are more secure, the common cases aren't significantly more secure than 3 × 3 patterns, suggesting that the expanded grid sizes aren't as beneficial for security as you might imagine.

This work shows that the impact of humans on security systems can be severe. Despite the fact that there are trillions of possible 4 × 4 patterns, the same habits of 3 × 3 patterns persist, greatly weakening the security system. As a larger trend, we as a community need to look for methods that improve security not just through increased choices, but with better directions and instructions. Users don't understand the power of computers to crack passwords, but there are straightforward and simple procedures for improving security — for example, simply picking a pattern that doesn't start in the upper left makes a huge impact. In the same way, for other systems, we need to consider the impact of humans on security systems; humans are often the weakest link.

## Acknowledgments

This work was supported in part by the US Office of Naval Research and the US National Security Agency, and we were assisted by Jeanne Luning-Prak, Flynn Wolf, and Rida Bazzi.

## References

1. R. Biddle, S. Chiasson, and P.C. Van Oorschot, "Graphical Passwords: Learning from the First Twelve Years," *ACM Computing Surveys*, vol. 44, no. 4, 2012, article no. 19.
2. S. Egelman et al., "Are You Ready to Lock?" *Proc. 2014 ACM SIGSAC Conf. Computer and Comm. Security*, 2014, pp. 750–761.
3. S. Uellenbeck et al., "Quantifying the Security of Graphical Passwords: The Case of Android Unlock Patterns," *Proc. 2013 ACM SIGSAC Conf. Computer & Comm. Security*, 2013, pp. 161–172.
4. A.J. Aviv, D. Budzitowski, and R. Kuber, "Is Bigger Better? Comparing User-Generated Passwords on 3 × 3 vs. 4 × 4 Grid Sizes for Android's Pattern Unlock," *Proc. 31st Ann. Computer Security Applications Conf.*, 2015, pp. 301–310.
5. A.J. Aviv et al., "Smudge Attacks on Smartphone Touch Screens," *Proc. 2010 Workshop on Offensive Technology*, 2010, pp. 1–7.
6. A.J. Aviv, J. Maguire, and J.L. Prak, "Analyzing the Impact of Collection Methods and Demographics for Android's Pattern Unlock," *Proc. Usable Security Workshop*, 2016; www.usna.edu/Users/cs/aviv/papers/aviv-usec16.pdf.
7. A.J. Aviv et al., "Practicality of Accelerometer Side Channels on Smartphones," *Proc. 28th Ann. Computer Security Applications Conf.*, 2012, pp. 41–50.
8. A. Adams and M.A. Sasse, "Users Are Not the Enemy," *Comm. ACM*, vol. 42, no. 12, 1999, pp. 40–46.
9. P.G. Kelley et al., "Guess Again (and Again and Again): Measuring Password Strength by Simulating Password-Cracking Algorithms," *Proc. IEE Symp. Security and Privacy*, 2012, pp. 523–537.
10. J. Bonneau, "The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords," *Proc. 2012 IEEE Symp. Security and Privacy*, 2012, pp. 538–552.
11. Y. Song et al., "On the Effectiveness of Pattern Lock Strength Meters: Measuring the Strength of Real World Pattern Locks," *Proc. 33rd Ann. ACM Conf. Human Factors in Computing Systems*, 2015.
12. H. Kim and J.H. Huh, "PIN Selection Policies: Are They Really Effective?" *Computers & Security*, vol. 31, no. 4, 2012, pp. 484–496.

**Adam J. Aviv** is an assistant professor of computer science at the United States Naval Academy in Annapolis, Maryland. His research interests include usability and security, particularly related to mobile devices. Aviv has a PhD in computer and information science from the University of Pennsylvania. Contact him at aviv@usna.edu.

**Ravi Kuber** is an associate professor of information systems at the University of Maryland, Baltimore County. He's worked extensively in accessibility research and computer-human interaction. Kuber has a PhD in information systems from Queen's University Belfast. Contact him at rkuber@umbc.edu.

**Devon Budzitowski** is a lieutenant (junior grade) in the US Navy. He's working on his MSE in computer science at the Naval Postgraduate school in Monterey, California. Budzitowski has a BS in computer science from the US Naval Academy. Contact him at debudzit@nps.edu.