On the In-Accuracy and Influence of Android Pattern Strength Meters

Work in Progress

Maximilian Golla
Ruhr University Bochum
maximilian.golla@rub.de

Jan Rimkus
Ruhr University Bochum
jan.rimkus@rub.de

Adam J. Aviv
United States Naval Academy
aviv@usna.edu

Markus Dürmuth
Ruhr University Bochum
markus.duermuth@rub.de

***Abstract*—A common method for helping users select stronger authentication secrets, e.g., passwords, is to deploy a visual strength meter that provides feedback to the user while performing password selection. Recent work considered the accuracy of strength meters for passwords, but there is limited work on understanding the accuracy of strength meters for other knowledge-based authentication types, particularly Android's graphical pattern unlock, despite there being multiple strength meters proposed for patterns in the literature. In this work, we present a preliminary analysis of the accuracy of strength meters for Android patterns, applying the same set of techniques from previous work. Using datasets of patterns collected in several user studies as a baseline, we compare strength meter estimations using weighted Spearman correlation. Overall, we find that strength estimations based on visual properties of the patterns (such as length, intersections, overlapping nodes, and similar) provide strength estimations that have low correlation with the real guessability of Android patterns. Motivated by these findings, we describe a set of research questions and experiments that are in progress that question whether the accuracy of a meter should even be the driving factor for nudging users to more secure choices.**

I. Introduction

Strength meters are commonplace during password selection, and studies have shown that user choice is influenced by them [40]. Sometimes, users are forced to achieve a high meter score, especially for high-value accounts [20]. Some meter designs take into account estimations of strength based on our understanding of how users select passwords [47] or provide feedback for improving user choice [38]. However, many password meters, particularly those currently deployed across the Web, do not well relate with our current understanding of password strength [42], [9], which considers large guessing attacks where the attacker has some knowledge of likely passwords. Meanwhile, most meters are prescriptive [17], e. g., only considering if a password is of sufficient length with the appropriate number of special, upper and lower case letters, which does not correlate well with guessing strength [24]. Meter inaccuracy may even cause more harm than good by providing users with a false sense of security [41], [39].

In the mobile authentication domain, strength meters have also been proposed for Android's graphical unlock scheme [1], [33], [31]. As a graphical knowledge-based authentication scheme, users select patterns by traversing a grid of 3x3 nodes without lifting, avoidance, or repetition (cf. Figure 1). All currently proposed pattern meters are based on visual properties, i. e., the shape of the pattern, rather than statistical features involved in the non-uniform selection of patterns by users, e. g., users tend to start in the upper left and end in the lower right. A number of studies have shown that human pattern choices are far simpler to guess and predict than selecting uniformly from the set of all possible $389,112$ patterns [37], [3], [31], [36], [15], [45].

In this work in progress paper, we analyze the strength meters proposed for Android patterns. Using a large data set of patterns collected in a number of different studies (summarized by Aviv and Dürmuth [5]), we evaluate each strength meter under realistic guessing attack conditions. The large data set, to which we refer to as "All," consists of $4,637$ patterns that are merged from four different user studies. Based on their statistical properties, we divided the dataset into three groups consisting of "Weak," "Medium," and "Strong" patterns. Overall, we find that meters based on visual properties greatly misrepresent the strength of a pattern and have low correlation with the real strength. In contrast, we recommend using a data-driven approach by using Markov models trained on user-chosen patterns to build an accurate meter.

However, a well-tuned strength meter must also well influence users, and based on these results, we propose a new set of studies to understand better what makes a good strength meter design. While we consider the approach recently proposed by Ur et al. [38], which tells users what is wrong with their password and how to improve it, a good starting point, we think that due to the different attack surface of patterns (namely, being a throttled attack) and the mobile setting it might not be the right solution for this setting. We also acknowledge that even the presence of any strength meter, regardless of accuracy, may positively influence users to select patterns differently. On the other hand, a more obtuse, strictly-enforced blacklisting approach without any user interaction may lead to user frustration and might not increase security [25].

Instead, we argue that a pattern strength meter does not need to be accurate as long as it will drive users away from weak and ineffective authentication choices. Motivated by prior

Workshop on Usable Security (USEC) 2019
24 February 2019, San Diego, CA, USA
ISBN 1-891562-57-6
https://dx.doi.org/10.14722/usec.2019.23025
www.ndss-symposium.org

work in understanding perceptions of pattern strength [7], we will design a new, *trained placebo meter* that is tuned to best influence users. However, as the perception of security is not real security, we will also make use of a non-enforcing blacklisting approach to help users avoid selections of very common patterns. In summary, our paper consists of two parts:

(i) Completed: We implement and analyze the strength meters proposed for Android patterns and our proposal based on a Markov model. Using a large dataset of patterns collected in a number of different studies we measure the similarity between the various strength meter outputs and an ideal *reference*.
(ii) Work in progress: We plan to conduct a user study to evaluate a *trained placebo meter* that is tuned to best influence users by following users' perception of strength. While avoiding too common patterns via a *non-enforcing* blacklisting approach, we like to investigate the influence of strength meter accuracy.

II. Background and Related Work

Android unlock patterns remain a popular mobile device lock mechanism [44], [19], [26]. It was first introduced in 2008 and is based on the Pass-Go scheme [34]. While there exist some minor vendor specific variations, the patterns are typically "drawn" using 9 nodes arranged in a 3x3 grid as visualized in Figure 1. A valid pattern consists of 4 to 9 nodes, only straight lines are allowed, no node can be selected more than once, and all nodes along a path are connected (unless a node was selected before).

Android's graphical password has been studied in many contexts, including for security (e. g., smudges [8], [13], shoulder surfing [43], [4], [18], other side-channel attacks [50], [14], [48], [49]), user choice [37], [1], [28], selection aids [36], [15], and under modifications [3], [35]. As mentioned, there has been a number of proposals for pattern strength meters [1], [33], [31], which we will describe in more detail in the following section. From this prior work on Android patterns, it is known that users tend to select non-uniformly in the space of the possible 389,112 possible patterns. There exists a common set of preferences, e. g., starting in the upper left and ending in the lower right, that make user-selected patterns highly predictable, repetitive, and easily guessed using methods such as Markov models [37]. Even then, if an attacker were to guess only the set of most common patterns, such as the **Z**-shaped pattern (as depicted in Figure 1), the attacker would be successful in many circumstances.

These facts motivated the design of strength meters for Android patterns. However, all of these designs are based on visual features (e. g., the length, or intersections of a pattern) rather than statistical properties; as a result, the accuracy and usefulness of these meters have been questioned. Closest to our work in analyzing the accuracy of pattern meters is a poster presented by Heidt and Aviv [27] in 2016. There, the authors compared the strength reported by meters measured across all patterns to a data-driven estimation of strength based on Markov modeling. Here, we expand greatly on this effort by adopting the techniques of Golla and Dürmuth [24] in evaluating the accuracy of text-based password strength meters. These methods are described in detail in the later sections.

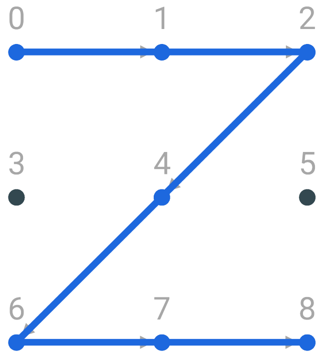


Fig. 1. Example **Z**-shaped Android unlock pattern `0.1.2.4.6.7.8`. Like 42 % of all analyzed human-chosen patterns, this pattern starts in the upper left corner and continues over the nodes `1` and `2` and ends in `8`. (The numbers and small arrows are added only for illustrative purposes.)

III. Preliminaries

In this section, we present three strength meters proposed for Android unlock patterns. Additionally, we describe our attacker model and the four different datasets of patterns used in our evaluation.

A. Pattern Strength Meters

We implemented three Android unlock patterns strength meters: Andriotis et al. [1], Sun et al. [33], and Song et al. [31]. Each of the pattern meters is referred to by the first author's last name. Each of these meters is based on different formulas and slightly different visual features; each will be described in the context of the meter. For consistency, we use the term *node* to refer to a contact point in the grid, and the length of a pattern is defined as the total number of *nodes* used in its definition. When referring to specific nodes in the grid, we use the numeric labeling, starting with `0` in the upper left and ending with `8` in the lower right, as depicted in Figure 1. A pattern P is defined as a sequence of nodes, e.g., `0.1.2.4.6.7.8` for the **Z**-shaped pattern. We use cardinality to define the length of a pattern, like $|P|$, which is the total number of nodes needed to draw the pattern. For example, the **Z**-shaped pattern is of length 7, and the **T**-shaped pattern of `7.4.1.0.2` is of length 5. Additionally, we use the term *segment* to refer to connected lines of the pattern. For example, the **Z**-shaped pattern has six segments, while the **T**-shaped pattern only has four segments.

1) Andriotis Meter: In 2014, Andriotis et al. [1] proposed an Android unlock pattern strength meter. The heuristic-driven meter is based on five visual properties: starting node, length, direction changes, knight moves, and the existence of overlapping nodes.

1A: *Starting Node* s: $s = 1$ if the starting node is *not* the top-left corner (node `0`), else $s = 0$.
1B: *Minimum Length* l': $l' = 0$ if the length $l = |P|$ of the pattern is five or shorter, else $l' = |P| - 5$.
1C: *Direction Changes* c: $c = 1$ if the number of direction changes (when three consecutive nodes are not on a straight line but form an angle different from 180 degrees) is at least 2, else $c = 0$. (For example, `2.4.3.5` has two direction changes).

1D: *Knight Moves* k: the number k of segments with Euclidean distance $\sqrt{5}$ (e. g., `0.7`, or `1.8`), like the knight moves in chess.

1E: *Overlapping Nodes* o: the number o of nodes that are retraced (e. g., the pattern `1.4.0.2` has one overlapping node, namely `1` is retraced when connecting `0` with `2`).

The score $S_{Andriotis}$ (ID: 1F/1G) is then defined as:

$$S_{Andriotis} = s + l' + c + k + o$$

Quantization (Q3): The score $S_{Andriotis}$ is divided into 3 strength categories: Weak ($0 \leq score \leq 1$), Medium ($score = 2$), and Strong ($score \geq 3$).

2) Sun Meter: Also in 2014, Sun et al. [33] proposed the following strength meter construction. Again, the meter is based on four visual properties: length, physical length, intersections, and overlaps. The authors motivated their construction, by following (the nowadays outdated) NIST's SP 800-63-2 [12] to calculate the entropy for a *randomly* selected password, which makes length the overall driving factor for strength estimation.

2A: *Length* l: is the length $|P|$ of the pattern.

2B: *Physical Length* l^e: is the Euclidean distance of the pattern (drawn on a unit grid).

2C: *Intersections* i: is the number of intersections, where segments cross each other. For example, the pattern `4.0.1.3` has one intersection where `4.0` crosses `1.3`. More evolved is the pattern `1.4.5.3`, which has one intersection where `1.4` touches `5.3` in node `4`.

2D: *Overlapping Segments* o: the number o of segments that are retraced. For example, the pattern `1.4.0.2` has no overlapping segments, but the **T**-shaped pattern `7.4.1.0.2` has one overlapping segment, namely `1.0` which is retraced when connecting node `0` with node `2`.

The score S_{Sun} (ID: 2E/2F) is defined as

$$S_{Sun} = l \cdot \log(l^e + i + o)$$

Quantization (Q5): The normalized score S_{Sun} is divided into 5 equal intervals that correspond to the strength categories: Very Weak, Weak, Medium, Strong, and Very Strong.

3) Song Meter: In 2015, Song et al. [31] proposed another meter based on visual properties. In contrast to previous work, their proposal introduces weights that are adjustable and can be used to change the importance of different visual features.

The meter is defined as follows:

3A: *Length in Maximum Norm* l^∞: is the length of the pattern wrt. the maximum norm (where the distance between two nodes $(x, y), (x', y')$ is $\max\{|x - x'|, |y - y'|\}$).

3B: *Intersections (Restricted)* i: is the number of intersections, where segments cross each other but restricted by not counting intersections that occur at the start or the end of a segment. Following this interpretation, the aforementioned pattern `1.4.5.3` has no intersections, as the "touch" happens *at the end* of the `1.4` segment.

3C: *Ratio of Non-Repeated Segments* n: is described loosely by Song et al. as "the number of times a segment does not appear in the longest repeated sub-pattern" to "the total number of segments in an input pattern p." We understand this to imply how many uniquely directed segments (considering only the *absolute* direction) appear in the pattern compared to the total segments. The **Z**-shaped pattern has two uniquely directed segments, the *horizontal* (`0.1;1.2;6.7;7.8`) and the *diagonal* uniquely directed segment (`2.4;4.6`). Furthermore, it has six segments in total resulting in a ratio of $2/6 = 1/3$.

The score S_{Song} (ID: 3D/3E) is defined as

$$S_{Song} = w_L \cdot \frac{l^\infty}{15} + w_N \cdot n + w_I \cdot \frac{\min\{i, 5\}}{5}.$$

After conducting a 101 participants user study, Song et al. suggested to use the following weights: $w_L = 0.81$, $w_N = 0.04$, and $w_I = 0.15$. Sadly, their description of the meter is not complete, and leaves some room for interpretation around the "Ratio of Non-Repeated Segments," leading to the interpretation presented above, even after multiple attempts to get more precise descriptions from the authors. Thus, our implementation of their meter follows our interpretation based on their description [31] and related work [27].

Quantization (Q3): The normalized score S_{Song} is divided into 3 strength categories: Weak (0.00–0.40), Medium (0.41–0.56), and Strong (0.57–1.00).

TABLE I. ANDROID GATEKEEPER: RATE-LIMITING

	Accumulated Waiting Time	
Guesses	**Android 6**	**Android 7, 8, 9**
1 guess	0 s	0 s
3 guesses	0 s	0 s
10 guesses	30 s	30 s
30 guesses	10 m 30 s	10 m 30 s
100 guesses	45 m 30 s	10 h 45 m 30 s
200 guesses	1 h 35 m 30 s	67 d 2 h 45 m 30 s
300 guesses	2 h 25 m 30 s	167 d 2 h 45 m 30 s

B. Threat Model

We consider a throttled guessing attack [10]. In such an attack, the n most common secrets are guessed in decreasing order of success. It follows the idea that an attacker will always guess the most common secrets first to maximize the overall success probability of the attack. Currently, in contrast to passwords, datasets of Android unlock patterns are not easy to get. Obtaining a ranking from frequent to infrequent patterns is even more difficult. Depending on the OS version, Android implements guessing throttling which rate-limits the number of guesses an attacker can make on the device before timeouts and, eventual, lockout. The accumulated time following timeouts at various guess numbers are shown in Table I. For Android 6 (which has the highest market share of 21.6 % [32] currently), 200 guesses is a reasonable assumption for a throttled attacker, especially considering one of our datasets has 199 example patterns. For the more modern Android 7 through 9 (accumulated market share of 48.5 % [32]) we consider up to 100 guesses (10 h 45 m 30 s [2]) to be reasonable.

One instance that we evaluate requires a more advanced attacker in possession of the top 20 patterns of the respective distribution. Whether or not such an attacker is a realistic assumption is open to speculation. It is known that the pattern distribution is significantly influenced by underlying demographic factors such as age, gender, and experience in IT [28],

TABLE II. DATASETS: GUESSING DIFFICULTY

						Throttled Guessing (Success in %)					
		Dataset	**Samples**	**Unique**	**∅ Length**	H_∞	λ_1	λ_3	λ_{10}	λ_{30}	λ_{100}
3x3 Patterns	**Strong**	ABK-MP-adv-defensive [3]	113	105	6.11	5.82 bits	1.77	5.31	15.93	33.63	95.58
		UDWH-defensive [37]	106	103	6.55	5.73 bits	1.89	5.66	12.26	31.13	97.17
		Combined: Strong	**219**	**199**	**6.32**	**6.19 bits**	**1.37**	**3.65**	**10.05**	**22.83**	**54.79**
	Medium	LDR-banking [28]	837	554	5.92	5.62 bits	2.03	5.73	11.59	21.98	42.41
		ZEBOdLAH [45]	507	350	5.03	5.40 bits	2.37	6.51	14.79	28.60	50.69
		ABK-MP-adv-offensive [3]	378	248	6.34	5.39 bits	2.38	7.14	17.20	35.19	60.85
		ABK-MP-self [3]	440	298	6.05	4.69 bits	3.86	8.18	17.73	32.05	55.00
		Combined: Medium	**2,162**	**1,067**	**5.81**	**5.79 bits**	**1.80**	**4.86**	**11.24**	**21.88**	**41.26**
	Weak	LDR-shopping [28]	841	442	5.54	5.02 bits	3.09	8.44	19.62	34.01	57.07
		LDR-unlock [28]	842	419	5.40	4.96 bits	3.21	8.55	17.46	31.95	57.60
		UDWH-offensive [37]	573	339	6.30	4.52 bits	4.36	10.12	19.02	35.60	58.29
		Combined: Weak	**2,256**	**886**	**5.68**	**5.05 bits**	**3.01**	**7.80**	**16.98**	**30.90**	**53.06**
	All	**Combined: All**	**4,637**	**1,635**	**5.77**	**5.61 bits**	**2.05**	**5.95**	**13.56**	**24.93**	**45.03**
		Uniform	389,112	389,112	7.97	18.57 bits	0.00	0.00	0.00	0.01	0.03
				Comparison							
PINs	**4-digit**	iPhone (Amitay) [11]	204,432	9,888	4.00	4.52 bits	4.35	9.24	14.45	20.63	29.27
		RockYou	1,780,587	10,000	4.00	4.75 bits	3.72	8.04	16.63	29.66	40.12
		Uniform	10,000	10,000	4.00	13.29 bits	0.01	0.03	0.10	0.30	1.00
	6-digit	RockYou	2,758,490	448,185	6.00	3.10 bits	11.69	12.76	14.77	17.33	19.95
		Uniform	1,000,000	1,000,000	6.00	19.93 bits	0.00	0.00	0.00	0.00	0.01
PW	**Mobile**	3class8MM [29]	273	273	10.36	8.09 bits	0.37	1.10	3.66	10.99	36.63
		3class12MM [29]	248	248	13.65	7.95 bits	0.40	1.21	4.03	12.10	40.32

Min-entropy (H_∞)**:** Is a worst-case metric for user-chosen secrets, considering an attacker who only guesses the most likely secret before giving up.
β**-success-rate:** Measures the expected guessing success for an adversary limited to β-guesses per account (λ_1 = 1 guess).

and potentially more such as cultural background and writing style [6]. We think a reasonably invested attacker might have access to a *generic* pattern dataset and its top 20 patterns, but most likely is not able to determine the exact order of the most common 20 patterns of the respective population.

C. Datasets

To measure the accuracy of the strength meter proposals (cf. Section III-A), we selected four datasets. The datasets contain patterns collected in various user studies by:

- Aviv et. al [3] (ABK-MP) in 2015
- Løge et. al [28] (LDR) in 2016
- Uellenbeck et. al [37] (UDWH) in 2013
- Von Zezschwitz et al. [45] (ZEBOdLAH) in 2016

An overview of the datasets is given in Table II. A more detailed description on how the data was collected, their statistical and visual properties, and their indented use case is given by Aviv and Dürmuth [5]. While a larger corpus of studies exists [5], only a minority shared their datasets or collected variants of the classical patterns that are not comparable. Overall, we evaluated $4,637$ human-chosen Android unlock patterns.

We report the throttled guessing resistance in terms of β-success-rate and Min-entropy (H_∞). The β-success-rate measures the expected guessing success for an adversary limited to β-guesses per account (λ_1 = 1 guess). The Min-entropy is a worst-case metric for user-chosen secrets, considering an attacker who only guesses the most likely secret before giving up. An introduction to the two metrics is given by Bonneau [9] and Wang et al. [46].

The $4,637$ patterns are merged from the four different user studies, and we will refer to them as "All" in the following sections. Based on their statistical properties we sorted the datasets from the individual studies into three groups, consisting of "Weak," "Medium," and "Strong" patterns, similar to what was done by Aviv and Dürmuth [5]. Due to the limited number of samples, guessing 100 patterns (in perfect order), for the two individual datasets ABK-MP-adv-defensive and UDWH-defensive results in very high λ_{100} values. However, the merge of the datasets consists of 199 unique samples, explaining the very different (albeit more realistic) λ_{100} value reported for the "Strong" dataset.

To better understand the overall level of protection offered by Android unlock patterns, we added statistics for 4-digit [11] and 6-digit [46] PINs and passwords that were created on mobile devices [29] as alternative mobile authentication schemes. For 4-digit PINs, we used Daniel Amitay's iPhone PIN dataset [11]. As there is no real-world 6-digit PIN data available, we contacted Wang et al. [46] to obtain a copy of their artificially crafted 6-digit PIN dataset. Unfortunately, they were not allowed to share their PINs with us. However, as their dataset has been generated by extracting PINs from password leaks, they recommended us to do the same. By following their PIN extraction method [46], we extracted 4 and 6-digit PINs from the RockYou password leak [16]. The β-success-rate and Min-entropy numbers of our extracted data are the same as the numbers reported by Wang et al. (requires the conversion from bits to percentages). However, one must be careful in drawing conclusions from the 6-digit artificial dataset, as the extracted data may not follow the same distribution as "normal" 6-digit PINs. For example, we observed that especially "123456" (the most common PIN) is overrepresented in this generated dataset (the PIN is up to 20x more frequent) resulting in a relatively high λ_1 value.

Besides PINs, passwords created on mobile devices are another common knowledge-based authentication mechanism. In 2016, Melicher et al. [29] studied the strength and usability of passwords that were created on mobile devices. We report the β-success-rate and Min-entropy for two conditions they studied. While both datasets of passwords were created and used on mobile devices (MM), they differ by their password composition policy. The first, 3class8MM, loosely follows the nowadays outdated NIST recommendation [12], by requiring at least 8 characters and at least three character classes. Furthermore, Melicher et al. tested a more usable and more secure alternative that requires at least 12 instead of 8 characters and also at least three character classes, called 3class12MM.

IV. Markov Models Strength Estimation

In this section, we outline how to calculate pattern strength based on Markov models, as a form of a data-driven meter.

To estimate the strength of patterns, Markov models have been used by Aviv et al. [3], Uellenbeck et al. [37], and Løge et al. [28]. Furthermore, the authors of pattern-creation guiding systems like SysPal [15] and TinPal [36] and alternative layouts like Pass-O [35] used Markov models to evaluate their proposals.

Markov models are based on the observation that subsequent tokens, such as letters in normal text or nodes in the pattern, are rarely independently chosen by humans and can often be quite accurately modeled based on a short history of tokens. For example, in English texts, the letter following a `t` is more likely to be an `h` than a `q`. For Android pattern unlock, a similar process occurs by which users are more likely to choose some transitions over others based on prior state, e. g., starting in the upper left, a user will more likely move to the right as opposed to the center node.

Using these observations, one can construct an n-gram Markov model that estimates the likelihood of a pattern by considering the probability of a transition from the previous $n-1$ nodes to the next node. For a given sequence of nodes for a pattern $p = \{c_1, \ldots, c_m\}$, an n-gram Markov model estimates its probability as

$$\begin{aligned} &P(c_1, \ldots, c_m) \\ &= P(c_1, \ldots, c_{n-1}) \cdot \prod_{i=n}^{m} P(c_i | c_{i-n+1}, \ldots, c_{i-1}). \end{aligned} \tag{1}$$

The required *initial probabilities* $P(c_1, \ldots, c_{n-1})$ and *transition probabilities* $P(c_n | c_1, \ldots, c_{n-1})$ can be determined empirically from the relative frequencies from training data. One commonly applies further post-processing to the raw frequencies: So-called *smoothing* tries to even out statistical effects in the transition matrix and initial probabilities. In particular, smoothing avoids relative frequencies of 0, as these would yield an overall probability of 0 regardless of the remaining probabilities. We use additive smoothing, i. e., the frequency of each n-gram is incremented by one.

To determine the strength of an individual pattern, we trained an n-gram Markov model using all $4,637$ available patterns with one instance of the pattern being measured left out. This process ensures the independence of training and evaluation sets while preserving information about pattern frequencies. For example, the pattern may be very frequent, but only a single instance is removed. While we have evaluated n-gram sizes from 2 to 5-grams (ID: 4A), we have decided to use 3-grams as the foundation for more advanced models based on the limited number of patterns in the training set and its use in previous works [37], [3].

We tested two advanced models (ID: 4B/4C): In the first (ID: 4B), we train individual Markov models per pattern length, motivated by previous work [23]. This approach typically yields better approximations and has been used before in the context of pattern security [3]. The latter variant (ID: 4C), follows the normal construction and training as described above, but additionally hard-codes the top 20 most frequent patterns of the respective datasets, i. e., "All," "Strong," "Medium," or "Weak." Guessing those 20 patterns in perfect order (without any approximation errors owed to the Markov model) improves accuracy. However, it requires an attacker that is in possession of this top 20 data, in contrast to an attacker that only trains a more generic Markov model based on all patterns available.

V. Meter Accuracy

We now present the results of measuring the accuracy of each of the strength meters (cf. Section III-A) and our proposed Markov model-based meter approach (cf. Section IV). From that, under our attacker model (cf. Section III-B), we show how the meters that rely on visual features inaccurately reflect the real strength of patterns.

A. Measuring Strength Meter Accuracy

The accuracy is a vital component for the overall performance of a strength meter, particularly for text-based passwords. Recent work by Golla and Dürmuth [24] systematically studied what constitutes a fair comparison of strength meters. The preferred method to measure the accuracy of a strength meter is to compare it to an ideal *reference*, measuring the *similarity* between the reference and the meter output. In their work, the authors suggest the *weighted Spearman's rank correlation coefficient* to be a useful candidate to measure the accuracy of a strength meter compared to the ideal reference.

In the following, we try to give an intuitive explanation of this coefficient. The "weighted" part means that some patterns are more important than others. For example, misjudging common patterns as "strong" is considered very bad, as such patterns are usually guessed early in an attack. In contrast, incorrectly over- or underestimating rare patterns is considered less problematic. A suitable weight could be the frequency of a pattern, as it is a direct indicator for the "importance" or commonness of a pattern. Furthermore, the "Spearman's rank" part means that the metric operates on the relative ranking instead of the direct meter output. Meters report strength values in different formats. Most common are home-grown scores, bits, probabilities, guess numbers, or the time required to guess the secret. To avoid any issues arising from the different output formats the metric operates on the relative ranking only. Thus, it is interested in whether pattern A is stronger than pattern B, but not by how much. Finally, the metric is a correlation coefficient thus it correlates two data points with each other.

In our case, the ground truth (sometimes called *reference*) is correlated with the meter output. We use the frequency (i. e., how often a pattern occurs) in a given dataset as the reference. Using the frequency, thus how likely a secret is, is the standard metric in throttled guessing scenarios. It follows the idea that an attacker will always guess in decreasing order of success, thus will guess very likely/common patterns first.

Implementation: All strength meters are implemented using the Kotlin programming language. This allows to run them natively on Android devices and enables us to transpile it to JavaScript to run them in a Web browser, too. The accuracy evaluation is implemented using R v3.5.1 (July 2018). For the weighted Spearman correlation we use the *wCorr* package[1]. All meter implementations and evaluation scripts are available online [30].

B. Results

An overview of the results is given in Table III. If the reported weighted Spearman correlation is close to 1.0 (high positive correlation), the meter accurately estimates pattern strength. If the correlation is close to 0.0 (no correlation), the meter is randomly guessing and does not estimate strength. Such meters have the potential to guide users in the wrong direction or confuse them. If the correlation is close to -1.0 (high negative correlation), the meter is accurate, but "strong" patterns are in fact "weak" and the other way round, misguiding users into choosing weak patterns instead of strong. In cases where the evaluated feature does not occur within the 200 most frequent patterns of the respective dataset, we display a hyphen (-).

In the following, we report the accuracy of various strength meters and their considered features by providing the total range from smallest to highest correlation value "(smallest-highest)" observed across all four datasets. Unsurprisingly all meters that rely on visual features fail to estimate strength accurately, given a throttled guessing scenario. Most promising are the visual features:

- 1A Starting Node (0.160-0.405),
- 1C Direction Changes (0.249-0.406).

Thus, especially the ID: 1F *Andriotis* meter that relies on both of those features performs most accurate among the meters considering only visual features. For the ID: 2E Sun meter we find results close to zero (-0.030-0.051). ID: 3D Song meter performs similarly inaccurate (-0.050-0.034). We found that both, Sun's as well as Song's meter do not estimate strength accurately in the considered throttled guessing scenario.

This result is not completely surprising, as similar observations have been made before in the context of LUDS-based password strength meters, counting the different character classes. A more intuitive explanation, about what is wrong with the meters, could be the following: Considering the aforementioned visually very complex pattern in **Z**-shape (`0.1.2.4.6.7.8`). The pattern is the second most common pattern in the datasets, thus has a guess number of 2. ID: 2E Sun and ID: 3D Song both assign it very high scores, due to its visual complexity that results in the guess numbers 161 and 153 out of a set consisting of the top 200 patterns. The same observation can be made in the password space, where the password "Password1" results in a very high score using a LUDS-based meter approach, while security experts know that it is a very weak password.

TABLE III. WEIGHTED SPEARMAN CORRELATION

ID	Meter	Quant.	Throttled Attacker - Top 200			
			All	**Strong**	**Medium**	**Weak**
1A	Starting Nodes		0.395	0.160	0.405	0.397
1B	Minimum Length		-0.034	0.018	-0.026	0.044
1C	Direction Changes		0.335	0.249	0.292	0.406
1D	Knight Moves		-	0.053	0.068	0.061
1E	Overlapping Nodes		0.139	0.110	0.089	0.148
1F	Andriotis		0.277	0.091	0.236	0.355
1G	Andriotis	Q3	0.121	0.132	0.111	0.214
2A	Length		-0.085	0.021	-0.038	-0.022
2B	Physical Length		-0.024	0.034	0.017	0.052
2C	Intersections		0.161	0.139	0.139	0.184
2D	Overlapping Segments		0.067	0.054	-	0.064
2E	Sun		-0.030	0.051	0.014	0.041
2F	Sun	Q5	-0.056	0.034	-0.032	0.018
3A	Length (Max. Norm)		-0.075	0.025	-0.028	-0.015
3B	Intersections (Rest.)		0.090	0.097	0.150	0.110
3C	Ratio NR Segments		0.166	0.089	0.135	0.110
3D	Song		-0.050	0.034	0.001	0.012
3E	Song	Q3	0.094	0.105	0.094	0.061
4A	Markov (2-grams)		0.313	0.176	0.325	0.396
4A	Markov (3-grams)		0.627	0.349	0.595	0.663
4A	Markov (4-grams)		0.684	0.371	0.645	0.709
4A	Markov (5-grams)		0.742	0.415	0.704	0.751
4B	Markov (3-grams - Length)		0.843	0.396	0.818	0.824
4C	Markov (3-grams - Top 20)		0.880	0.655	0.870	0.916

Quantization: Q3–Q5 = Number of bins, e. g., Q3 = [`Weak, Medium, Strong`]; **Weighted Spearman correlation:** 1.0 (high positive correlation); the meter accurately estimates pattern strength. 0.0 (no correlation); the meter is randomly guessing, and does not estimate strength. -1.0 (high negative correlation); the meter is accurate but "strong" patterns are in fact "weak" and the other way round.

In contrast, our results show the potential of using probabilistic approaches like Markov models to estimate the strength of a pattern. Our suggested generic Markov model (ID: 4A) reproduces the pattern ranking of the reference quite well (0.176-0.751). While one can observe an increasing accuracy with increasing n-gram size (2-grams: 0.176-0.396, 5-grams: 0.415-0.751), one needs to consider the sparse data situation and the risk of overfitting. Like previous work, we find the approach to train a separate model per pattern length (ID: 4B) to improve the accuracy (3-grams: 0.396-0.843). However, the performance degrades if not enough samples per length are available, as one can observe for the "Strong" dataset. As expected, hard-coding the top 20 patterns (ID: 4C) performs best (3-grams: 0.655-0.916) but requires a stronger attacker that is in possession of the top 20 patterns for each evaluated dataset.

The adverse effects of quantization, i. e., decreasing accuracy once the meter output is binned, for example into 3 bins (Q3) [`Weak, Medium, Strong`] has been found before [24]. The lower the number of bins, the more information in the relative ranking is lost; thus the weighted Spearman correlation degrades. This effect can be observed across all binning meters (ID: 1G, 2F, 3E).

[1] Package: wCorr (Weighted Correlations), Version 1.9.1, May 2017, https://cran.r-project.org/package=wCorr, as of January 15, 2019

VI. Work in Progress: Meter Influence

Next, we outline our current effort towards understanding the influence of strength meters on users' pattern choices.

A. Motivation

Considering the stringent rate-limiting (cf. Table I) deployed in modern Android versions 7+, and a reasonably determined throttled attacker, there is no need for patterns to repel much more than 100 guesses. Forcing users to create very strong patterns above this threshold, will lead to patterns falling into the so-called "don't care" region, as introduced by Florêncio et al. [21] in the context of password strength. Even worse, in contrast to passwords, there is no chance for a pattern to survive an unthrottled guessing attack, due to the small theoretical keyspace. Generally speaking, every pattern in this "don't care" region must be considered a waste of cognitive effort.

All three proposed pattern meters were evaluated in user studies [1], [33], [31]. For example, Sun et al. tested their meter with 81 participants and found: "*... the presence of a visual indicator of pattern strength did encourage users to create visually complex patterns.*" It seems that the mere presence of a strength indicator that follows the intuitive thought process is enough to motivate users. We can imagine that as long as the meter follows users' perception of strength [39], [41], it might be enough to nudge them toward choices outside the easily guessed threshold, but below the "don't care" chasm.

Based on this scenario we claim that while a high accuracy is important for meter developers, it certainly is not the driving factor for users. Instead, we like to explore whether a meter should engage more with users' perception of strength while still preventing too common, thus, very dangerous pattern choices. To this point, there is little known about user perception of pattern security. One study investigating this issue was conducted by Aviv and Fichter [7] in 2014. They identified the length (Euclidean distance) as one of the most influential visual features in user perception of pattern strength.

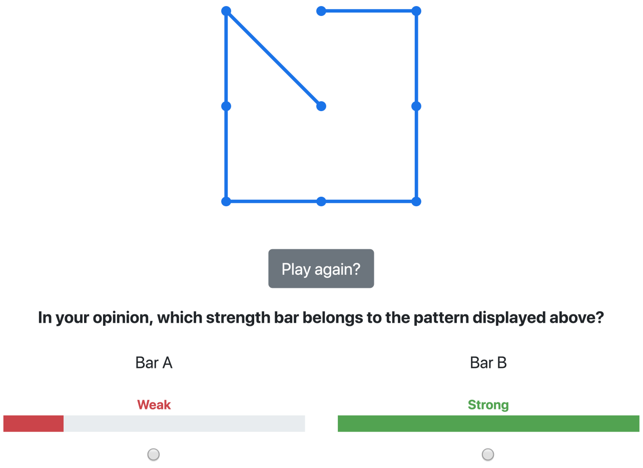
Based on those observations we developed the following questions that we like to answer by conducting a user study:

RQ1: How important is the accuracy of a strength meter to nudge users towards more secure patterns?

RQ2: Can a new strength meter that follows perceived instead of actual pattern strength, actively reduce the number of too common patterns, while not wasting users' cognitive effort in the "don't care" region?

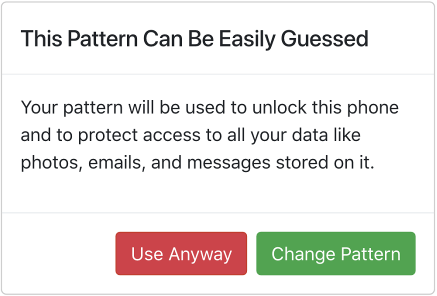
B. Methodology

Our prospective study design follows previous work by Aviv and Fichter [7] and Ur et al. [39]. In the study, users are presented with an animation that shows the creation of a pattern, as one can see in Figure 2. As patterns can be very complex and hard to follow given a static image, we use CSS animations to visualize the pattern creation process and provide a button to replay this animation. In contrast to previous work, we display two interactive strength meters below the pattern. Users are then asked to judge the strength of the pattern by deciding which of the two strength meter bars is more accurate.



In your opinion, which strength bar belongs to the pattern displayed above?

Fig. 2. Comparison of an accurate (left) and intuitive strength estimation (right). Participants need to judge which of the two strength estimates is more accurate. The pattern `1.2.5.8.7.6.3.0.4` is a common pattern, but typically outside the reach of throttled guessing attacks. Thus, the trained placebo meter should estimate it (based on its visual complexity) as "Strong."



This Pattern Can Be Easily Guessed

Your pattern will be used to unlock this phone and to protect access to all your data like photos, emails, and messages stored on it.

Fig. 3. Imagined warning dialog that tries to encourage users to reconsider their pattern choice, if a too common pattern is chosen. The dialog is based on Apple iOS's Passcode warning that implements a non-enforcing blacklist [22].

The two meters displayed in Figure 2, are very different. The first meter (left) is an accurate estimation of strength based on our Markov model-based proposal. The second meter (right) follows the intuitive idea of pattern strength of users based on the results from previous work [7] (which is mainly driven by visual complexity) but prevents too common choices like the **Z**-shape (`0.1.2.4.6.7.8`) pattern.

The idea of using this approach is to understand what is more influential for users: the highly accurate meter, or a trained placebo meter that behaves intuitively for users but prevents dangerous choices. After deciding which meter is more accurate, we will ask users about *why* they believe one meter gives more accurate feedback than the other. This way, we hope to learn more about their decision process that could help us to determine the importance of actual accuracy over perceived pattern strength accuracy.

To prevent users from choosing too weak patterns, we imagine a dialog (cf. Figure 3) similar to Apple iOS's Passcode warning that implements a non-enforcing blacklist [22]. At the same time, the trained placebo meter will estimate the strength of the pattern as "Too Weak," while showing an empty strength bar.

We believe that the approach of using a trained placebo meter is more intuitive than strength estimations and will become easier to comprehend for users. At the same time, the design promises the same security by preventing too common choices using a non-enforcing blacklist approach combined with a strength meter.

VII. Conclusion

In this work, we explored the accuracy of pattern strength meters using multiple datasets. Our results indicate that current proposals from the literature, which are solely based on visual features, do not estimate pattern strength and are a potentially dangerous substitute for statistical estimators, such as those based on Markov models. However, we acknowledge that the sheer presence of a meter, regardless of accuracy, is likely beneficial in the throttled guessing attacker model in which Android patterns exist. The primary goal should be to ensure that users do not select very common patterns, but are also not overburdened with attempting to select overly complex patterns.

To this end, we are in the process of designing a future study to investigate how vital meter accuracy is to nudge users towards more secure patterns. We will explore the use of a strength meter that follows perceived pattern strength instead of actual strength while maintaining security via a non-enforcing blacklist of too common patterns. It is our goal to actively reduce the number of patterns that are at risk in a throttled guessing attack while not wasting cognitive user effort.

References

[1] P. Andriotis, T. Tryfonas, and G. Oikonomou, "Complexity Metrics and User Strength Perceptions of the Pattern-Lock Graphical Authentication Method," in *Conference on Human Aspects of Information Security, Privacy and Trust*, ser. HAS '14. Heraklion, Crete, Greece: Springer, Jun. 2014, pp. 115–126.

[2] Android Open Source Project, "Android 9 – "Pie": GateKeeper – ComputeRetryTimeout Function," Feb. 2018, https://android.googlesource.com/platform/system/gatekeeper/+/pie-release/gatekeeper.cpp#253, as of January 15, 2019.

[3] A. J. Aviv, D. Budzitowski, and R. Kuber, "Is Bigger Better? Comparing User-Generated Passwords on 3x3 vs. 4x4 Grid Sizes for Android's Pattern Unlock," in *Annual Computer Security Applications Conference*, ser. ACSAC '15. Los Angeles, California, USA: ACM, Dec. 2015, pp. 301–310.

[4] A. J. Aviv, J. T. Davin, F. Wolf, and R. Kuber, "Towards Baselines for Shoulder Surfing on Mobile Authentication," in *Annual Conference on Computer Security Applications*, ser. ACSAC '17. Orlando, Florida, USA: ACM, Dec. 2017, pp. 486–498.

[5] A. J. Aviv and M. Dürmuth, "A Survey of Collection Methods and Cross-Data Set Comparison of Android Unlock Patterns," *CoRR*, vol. abs/1811.10548, pp. 1–20, Nov. 2018.

[6] A. J. Aviv, M. Dürmuth, and P. Gupta, "Position Paper: Measuring the Impact of Alphabet and Culture on Graphical Passwords," in *Who Are You?! Adventures in Authentication Workshop*, ser. WAY '16. Denver, Colorado, USA: USENIX, Jun. 2016.

[7] A. J. Aviv and D. Fichter, "Understanding Visual Perceptions of Usability and Security of Android's Graphical Password Pattern," in *Annual Computer Security Applications Conference*, ser. ACSAC '14. New Orleans, Louisiana, USA: ACM, Dec. 2014, pp. 286–295.

[8] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge Attacks on Smartphone Touch Screens," in *USENIX Workshop on Offensive Technologies*, ser. WOOT '10. Washington, District of Columbia, USA: USENIX, Aug. 2010, pp. 1–7.

[9] J. Bonneau, "The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords," in *IEEE Symposium on Security and Privacy*, ser. SP '12. San Jose, California, USA: IEEE, May 2012, pp. 538–552.

[10] J. Bonneau, C. Herley, P. C. Van Oorschot, and F. Stajano, "The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes," in *IEEE Symposium on Security and Privacy*, ser. SP '12. San Jose, California, USA: IEEE, May 2012, pp. 553–567.

[11] J. Bonneau, S. Preibusch, and R. Anderson, "A Birthday Present Every Eleven Wallets? The Security of Customer-Chosen Banking PINs," in *Financial Cryptography and Data Security*, ser. FC '12. Kralendijk, Bonaire: Springer, Feb. 2012, pp. 25–40.

[12] W. E. Burr, D. F. Dodson, and W. T. Polk, "Electronic Authentication Guideline: NIST Special Publication 800-63-2," Aug. 2013.

[13] S. Cha, S. Kwag, H. Kim, and J. H. Huh, "Boosting the Guessing Attack Performance on Android Lock Patterns with Smudge Attacks," in *ACM Asia Conference on Computer and Communications Security*, ser. ASIA CCS '17. Abu Dhabi, United Arab Emirates: ACM, Apr. 2017, pp. 313–326.

[14] P. Cheng, I. E. Bagci, U. Roedig, and J. Yan, "SonarSnoop: Active Acoustic Side-Channel Attacks," *CoRR*, vol. abs/1808.10250, pp. 1–13, Aug. 2018.

[15] G. Cho, J. H. Huh, J. Cho, S. Oh, Y. Song, and H. Kim, "SysPal: System-Guided Pattern Locks for Android," in *IEEE Symposium on Security and Privacy*, ser. SP '17. San Jose, California, USA: IEEE, May 2017, pp. 338–356.

[16] N. Cubrilovic, "RockYou Hack: From Bad To Worse," Dec. 2009, https://techcrunch.com/2009/12/14/rockyou-hack-security-myspace-facebook-passwords/, as of January 15, 2019.

[17] X. de Carné de Carnavalet and M. Mannan, "From Very Weak to Very Strong: Analyzing Password-Strength Meters," in *Symposium on Network and Distributed System Security*, ser. NDSS '14. San Diego, California, USA: ISOC, Feb. 2014.

[18] A. De Luca, M. Harbach, E. von Zezschwitz, M.-E. Maurer, B. E. Slawik, H. Hussmann, and M. Smith, "Now You See Me, Now You Don't: Protecting Smartphone Authentication from Shoulder Surfers," in *ACM Conference on Human Factors in Computing Systems*, ser. CHI '14. Toronto, Ontario, Canada: ACM, Apr. 2014, pp. 2937–2946.

[19] S. Egelman, S. Jain, R. S. Portnoff, K. Liao, S. Consolvo, and D. Wagner, "Are You Ready to Lock? Understanding User Motivations for Smartphone Locking Behaviors," in *ACM Conference on Computer and Communications Security*, ser. CCS '14. Scottsdale, Arizona, USA: ACM, Nov. 2014, pp. 750–761.

[20] S. Egelman, A. Sotirakopoulos, I. Muslukhov, K. Beznosov, and C. Herley, "Does My Password Go Up to Eleven?: The Impact of Password Meters on Password Selection," in *ACM Conference on Human Factors in Computing Systems*, ser. CHI '13. Paris, France: ACM, Apr. 2013, pp. 2379–2388.

[21] D. Florêncio, C. Herley, and P. C. Van Oorschot, "Pushing on String: The "Don't Care" Region of Password Strength," *Communications of the ACM*, vol. 59, no. 11, pp. 66–74, Oct. 2016.

[22] C. Gartenberg, "Kanye West's iPhone Passcode is 000000," Oct. 2018, https://www.theverge.com/tldr/2018/10/11/17964848/kanye-west-iphone-passcode-trump-iplane-apple-meeting, as of January 15, 2019.

[23] M. Golla, B. Beuscher, and M. Dürmuth, "On the Security of Cracking-Resistant Password Vaults," in *ACM Conference on Computer and Communications Security*, ser. CCS '16. Vienna, Austria: ACM, Oct. 2016, pp. 1230–1241.

[24] M. Golla and M. Dürmuth, "On the Accuracy of Password Strength Meters," in *ACM Conference on Computer and Communications Security*, ser. CCS '18. Toronto, Ontario, Canada: ACM, Oct. 2018, pp. 1567–1582.

[25] H. Habib, J. Colnago, W. Melicher, B. Ur, S. Segreti, L. Bauer, N. Christin, and L. Cranor, "Password Creation in the Presence of Blacklists," in *Workshop on Usable Security*, ser. USEC '17. San Diego, California, USA: ISOC, Feb. 2017.

[26] M. Harbach, E. von Zezschwitz, A. Fichtner, A. De Luca, and M. Smith, "It's a Hard Lock Life: A Field Study of Smartphone (Un)Locking

Behavior and Risk Perception," in *Symposium on Usable Privacy and Security*, ser. SOUPS '14. Menlo Park, California, USA: USENIX, Jul. 2014, pp. 213–230.

[27] S. Heidt and A. J. Aviv, "Poster: Refining Graphical Password Strength Meters for Android Phones," in *Symposium on Usable Privacy and Security*, ser. SOUPS '15. Ottawa, Ontario, Canada: USENIX, Jul. 2015.

[28] M. Løge, M. Dürmuth, and L. Røstad, "On User Choice for Android Unlock Patterns," in *European Workshop on Usable Security*, ser. EuroUSEC '16. Darmstadt, Germany: ISOC, Jul. 2016.

[29] W. Melicher, D. Kurilova, S. M. Segreti, P. Kalvani, R. Shay, B. Ur, L. Bauer, N. Christin, L. F. Cranor, and M. L. Mazurek, "Usability and Security of Text Passwords on Mobile Devices," in *ACM Conference on Human Factors in Computing Systems*, ser. CHI '16. Santa Clara, California, USA: ACM, May 2016, pp. 527–539.

[30] Rimkus, Jan, "APC: Android (Unlock) Pattern Classifier," Jan. 2019, https://github.com/RUB-SysSec/APC, as of January 15, 2019.

[31] Y. Song, G. Cho, S. Oh, H. Kim, and J. H. Huh, "On the Effectiveness of Pattern Lock Strength Meters: Measuring the Strength of Real World Pattern Locks," in *ACM Conference on Human Factors in Computing Systems*, ser. CHI '15. Seoul, Republic of Korea: ACM, Apr. 2015, pp. 2343–2352.

[32] Statista Inc., "Android Version Market Share Distribution Among Smartphone Owners," Sep. 2018, https://www.statista.com/statistics/271774/share-of-android-platforms-on-mobile-devices-with-android-os/, as of January 15, 2019.

[33] C. Sun, Y. Wang, and J. Zheng, "Dissecting Pattern Unlock: The Effect of Pattern Strength Meter on Pattern Selection," *Journal of Information Security and Applications*, vol. 19, no. 4–5, pp. 308–320, Nov. 2014.

[34] H. Tao and C. Adams, "Pass-Go: A Proposal to Improve the Usability of Graphical Passwords," *International Journal of Network Security*, vol. 7, no. 2, pp. 273–292, Sep. 2008.

[35] H. Tupsamudre, V. Banahatti, S. Lodha, and K. Vyas, "Pass-O: A Proposal to Improve the Security of Pattern Unlock Scheme," in *ACM Asia Conference on Computer and Communications Security*, ser. ASIA CCS '17. Abu Dhabi, United Arab Emirates: ACM, Apr. 2017, pp. 400–407.

[36] H. Tupsamudre, S. Vaddepalli, V. Banahatti, and S. Lodha, "TinPal: An Enhanced Interface for Pattern Locks," in *Workshop on Usable Security*, ser. USEC '18. San Diego, California, USA: ISOC, Feb. 2018.

[37] S. Uellenbeck, M. Dürmuth, C. Wolf, and T. Holz, "Quantifying the Security of Graphical Passwords: The Case of Android Unlock Patterns," in *ACM Conference on Computer and Communications Security*, ser. CCS '13. Berlin, Germany: ACM, Oct. 2013, pp. 161–172.

[38] B. Ur, F. Alfieri, M. Aung, L. Bauer, N. Christin, J. Colnago, L. F. Cranor, H. Dixon, P. E. Naeini, H. Habib, N. Johnson, and W. Melicher, "Design and Evaluation of a Data-Driven Password Meter," in *ACM Conference on Human Factors in Computing Systems*, ser. CHI '17. Denver, Colorado, USA: ACM, May 2017, pp. 3775–3786.

[39] B. Ur, J. Bees, S. M. Segreti, L. Bauer, N. Christin, and L. F. Cranor, "Do Users' Perceptions of Password Security Match Reality?" in *ACM Conference on Human Factors in Computing Systems*, ser. CHI '16. Santa Clara, California, USA: ACM, May 2016, pp. 3748–3760.

[40] B. Ur, P. G. Kelley, S. Komanduri, J. Lee, M. Maass, M. L. Mazurek, T. Passaro, R. Shay, T. Vidas, L. Bauer, N. Christin, and L. F. Cranor, "How Does Your Password Measure Up? The Effect of Strength Meters on Password Creation," in *USENIX Security Symposium*, ser. SSYM '12. Bellevue, Washington, USA: USENIX, Aug. 2012, pp. 65–80.

[41] B. Ur, F. Noma, J. Bees, S. M. Segreti, R. Shay, L. Bauer, N. Christin, and L. F. Cranor, ""I Added '!' at the End to Make It Secure": Observing Password Creation in the Lab," in *Symposium on Usable Privacy and Security*, ser. SOUPS '15. Ottawa, Ontario, Canada: USENIX, Jul. 2015, pp. 123–140.

[42] B. Ur, S. M. Segreti, L. Bauer, N. Christin, L. F. Cranor, S. Komanduri, D. Kurilova, M. L. Mazurek, W. Melicher, and R. Shay, "Measuring Real-World Accuracies and Biases in Modeling Password Guessability," in *USENIX Security Symposium*, ser. SSYM '15. Washington, District of Columbia, USA: USENIX, Aug. 2015, pp. 463–481.

[43] E. von Zezschwitz, A. De Luca, P. Janssen, and H. Hussmann, "Easy to Draw, but Hard to Trace?: On the Observability of Grid-based (Un)Lock Patterns," in *ACM Conference on Human Factors in Computing Systems*, ser. CHI '15. Seoul, Republic of Korea: ACM, Apr. 2015, pp. 2339–2342.

[44] E. von Zezschwitz, P. Dunphy, and A. De Luca, "Patterns in the Wild: A Field Study of the Usability of Pattern and PIN-based Authentication on Mobile Devices," in *Conference on Human-Computer Interaction with Mobile Devices and Services*, ser. MobileHCI '13. Munich, Germany: ACM, Aug. 2013, pp. 261–270.

[45] E. von Zezschwitz, M. Eiband, D. Buschek, S. Oberhuber, A. De Luca, F. Alt, and H. Hussmann, "On Quantifying the Effective Passsword Space of Grid-Based Unlock Gestures," in *Conference on Mobile and Ubiquitous Multimedia*, ser. MUM '16. Rovaniemi, Finland: ACM, Dec. 2016, pp. 201–212.

[46] D. Wang, Q. Gu, X. Huang, and P. Wang, "Understanding Human-Chosen PINs: Characteristics, Distribution and Security," in *ACM Asia Conference on Computer and Communications Security*, ser. ASIA CCS '17. Abu Dhabi, United Arab Emirates: ACM, Apr. 2017, pp. 372–385.

[47] D. L. Wheeler, "zxcvbn: Low-Budget Password Strength Estimation," in *USENIX Security Symposium*, ser. SSYM '16. Austin, Texas, USA: USENIX, Aug. 2016, pp. 157–173.

[48] G. Ye, Z. Tang, D. Fang, X. Chen, K. I. Kim, B. Taylor, and Z. Wang, "Cracking Android Pattern Lock in Five Attempts," in *Symposium on Network and Distributed System Security*, ser. NDSS '17. San Diego, California, USA: ISOC, Feb. 2017.

[49] G. Ye, Z. Tang, D. Fang, X. Chen, W. Wolff, A. J. Aviv, and Z. Wang, "A Video-based Attack for Android Pattern Lock," *ACM Transactions on Privacy and Security*, vol. 21, no. 4, pp. 19:1–19:31, Jul. 2018.

[50] M. Zhou, Q. Wang, J. Yang, Q. Li, F. Xiao, Z. Wang, and X. Chen, "PatternListener: Cracking Android Pattern Lock Using Acoustic Signals," in *ACM Conference on Computer and Communications Security*, ser. CCS '18. Toronto, Ontario, Canada: ACM, Oct. 2018, pp. 1775–1787.