Towards usable and secure graphical passwords for smartphones

Hyoungshick Kim

Sungkyunkwan University, CSIRO Data61

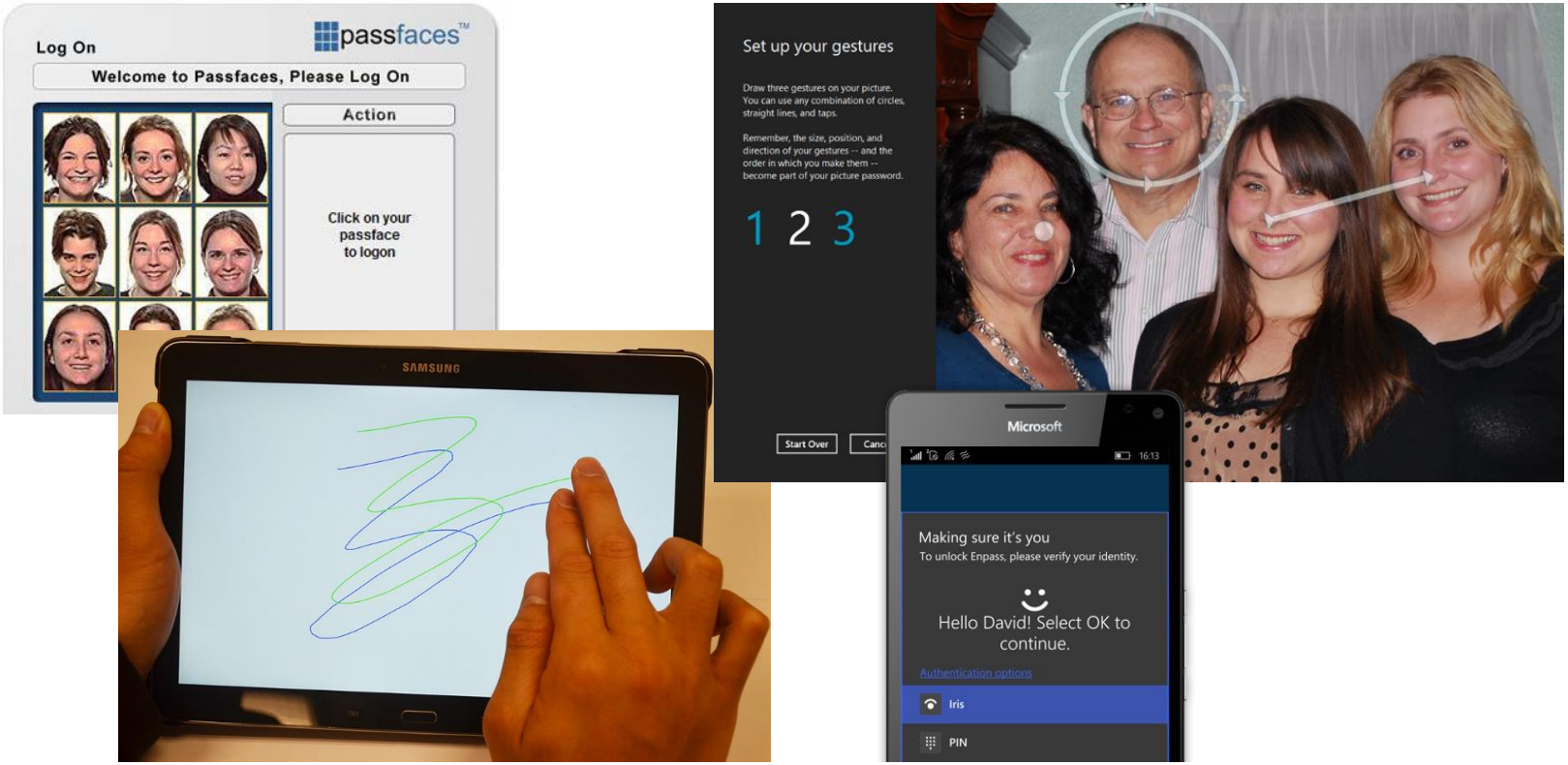
Difficult to type in mobile devices

Biometric Authentication

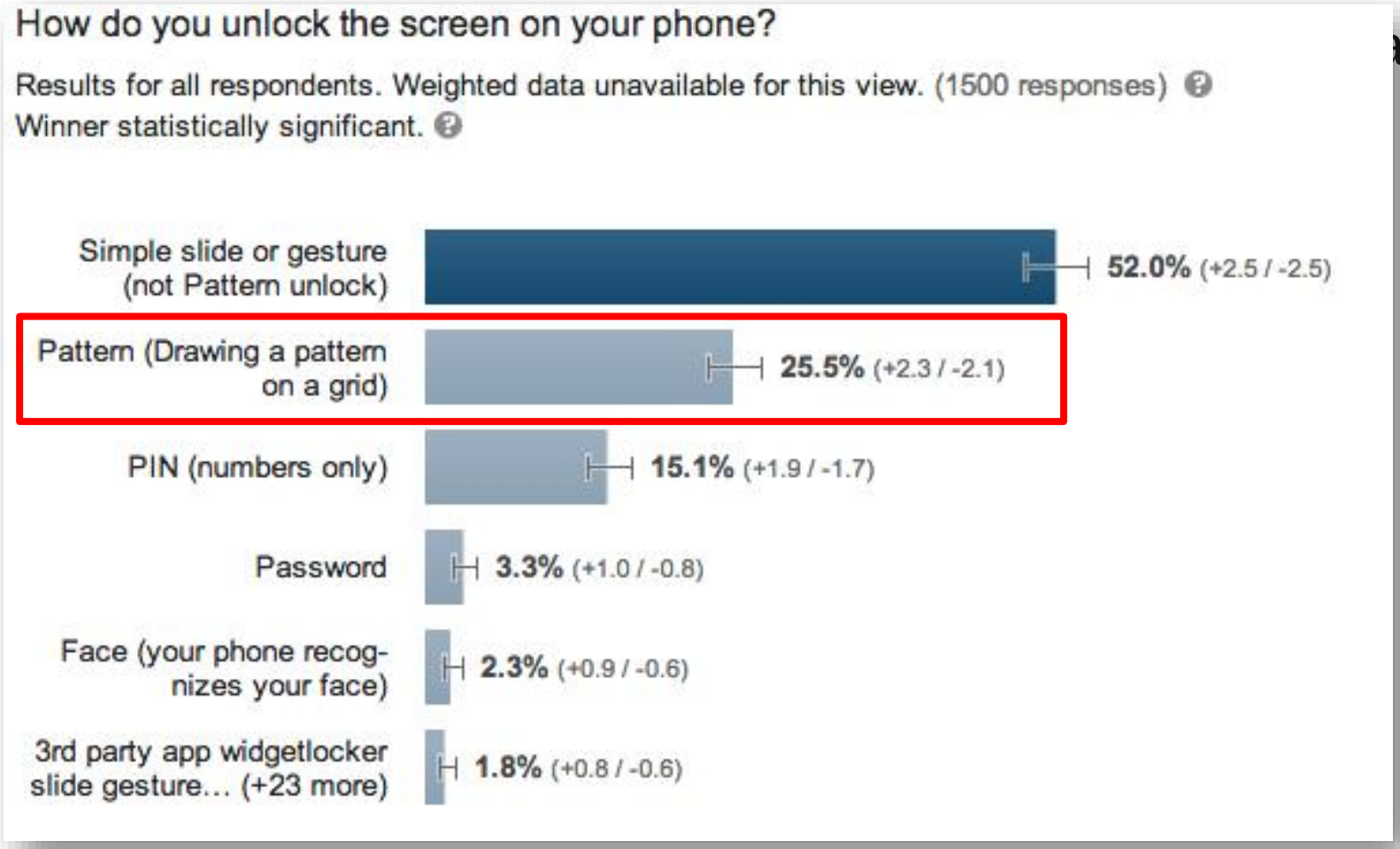
Password or PIN is still needed because biometric authentication schemes cannot sometimes work.

Graphical Passwords

- Many graphical password schemes have been introduced.

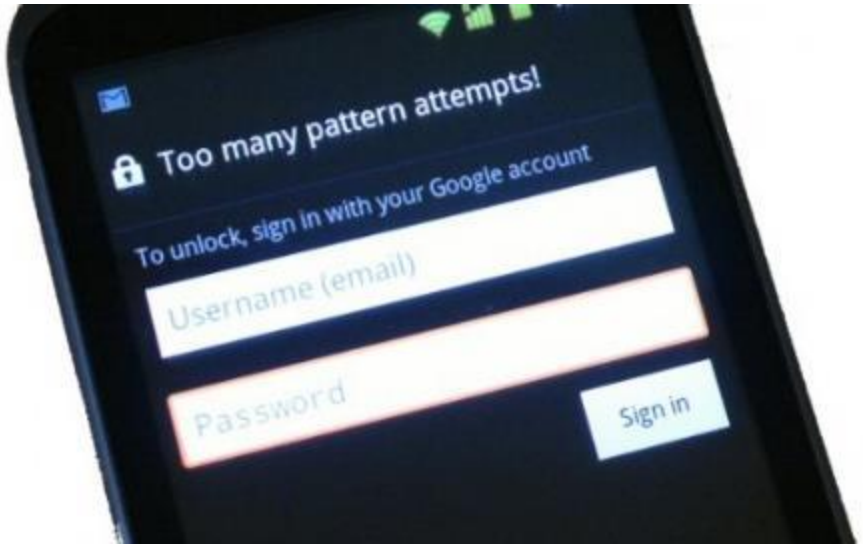


Android Pattern

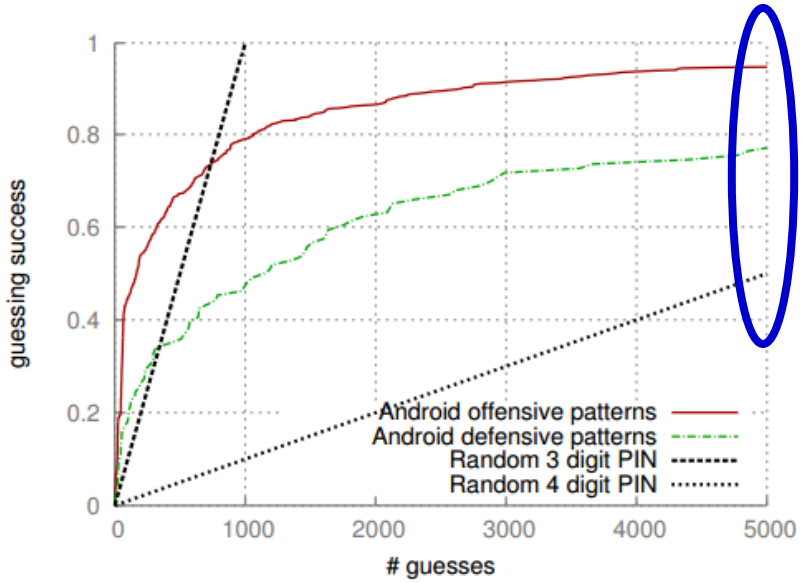


Threat Model for Patterns

- There are many known attacks against Android pattern locks.
  - Smudge attacks.
  - Sensor-based side channel attacks.
  - Shoulder surfing attacks.
  - Guessing attacks.
    with 20 consecutive guessing attempts allowed



Highly Guessable Patterns

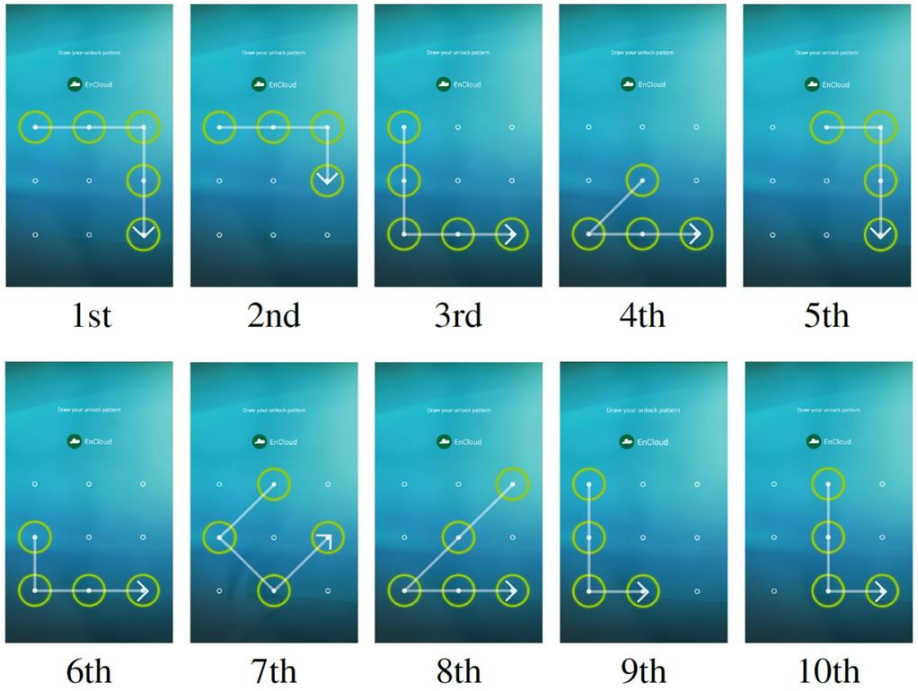


The guessing entropy of **user-chosen patterns is lower** than Random 4-digit PINs

Uellenbeck et al.,"Quantifying the security of graphical passwords: the case of android unlock patterns," **ACM CCS, 2013**.

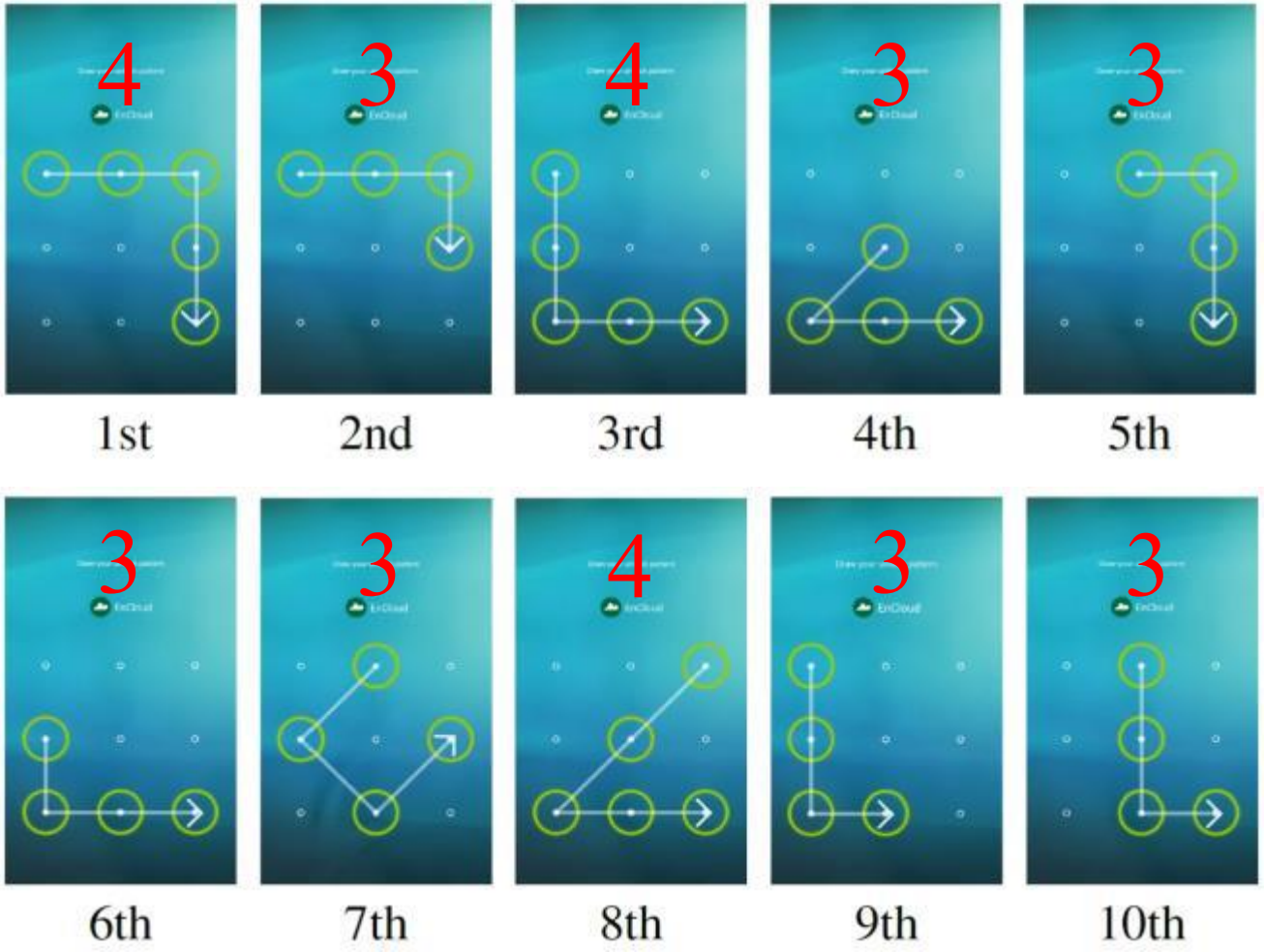
Top 10 Patterns

- The pattern distribution is highly skewed toward a small number of commonly used password.

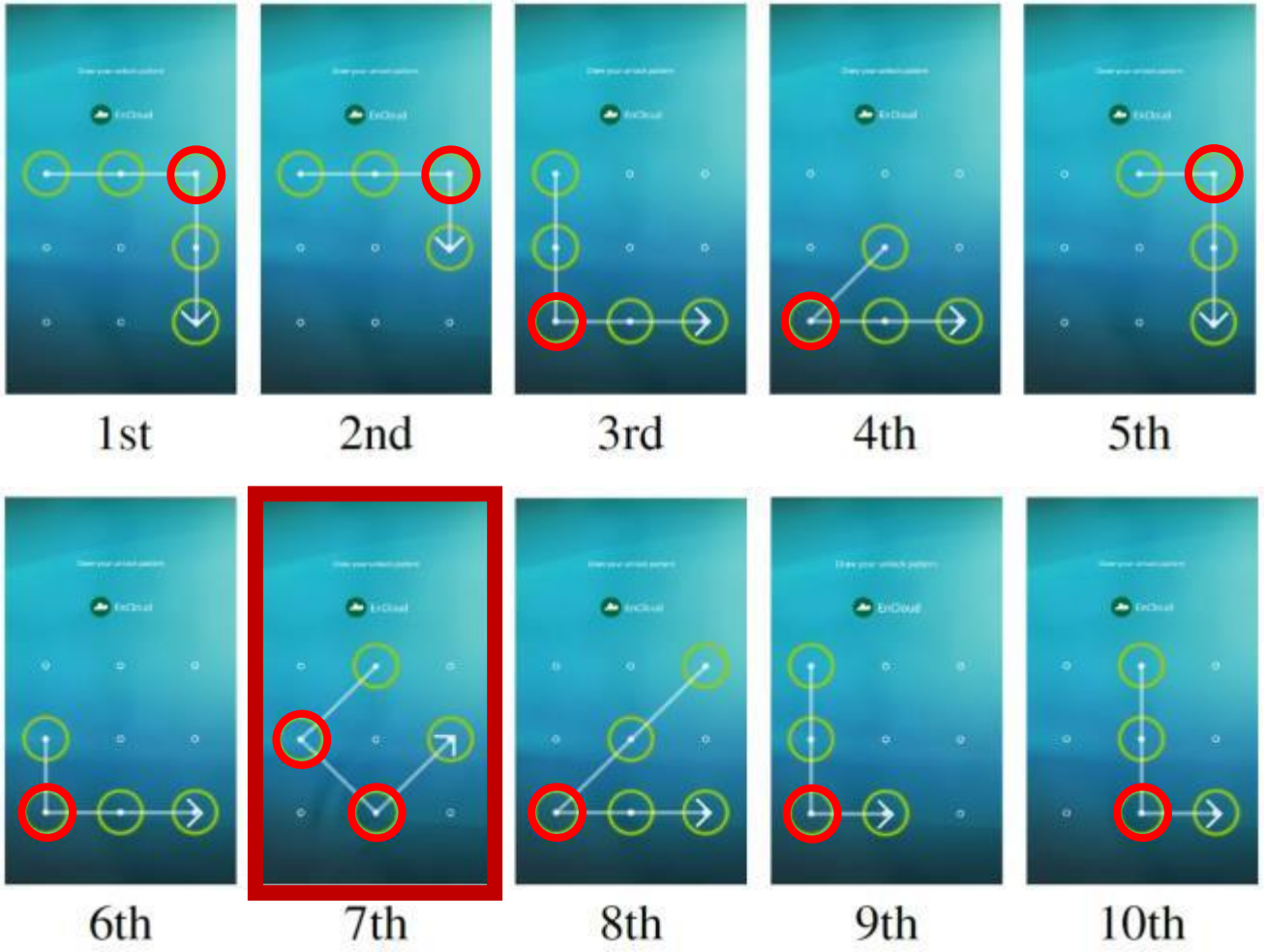


"On the Effectiveness of Pattern Lock Strength Meters: Measuring the Strength of Real World Pattern Locks," **ACM CHI, 2015.**

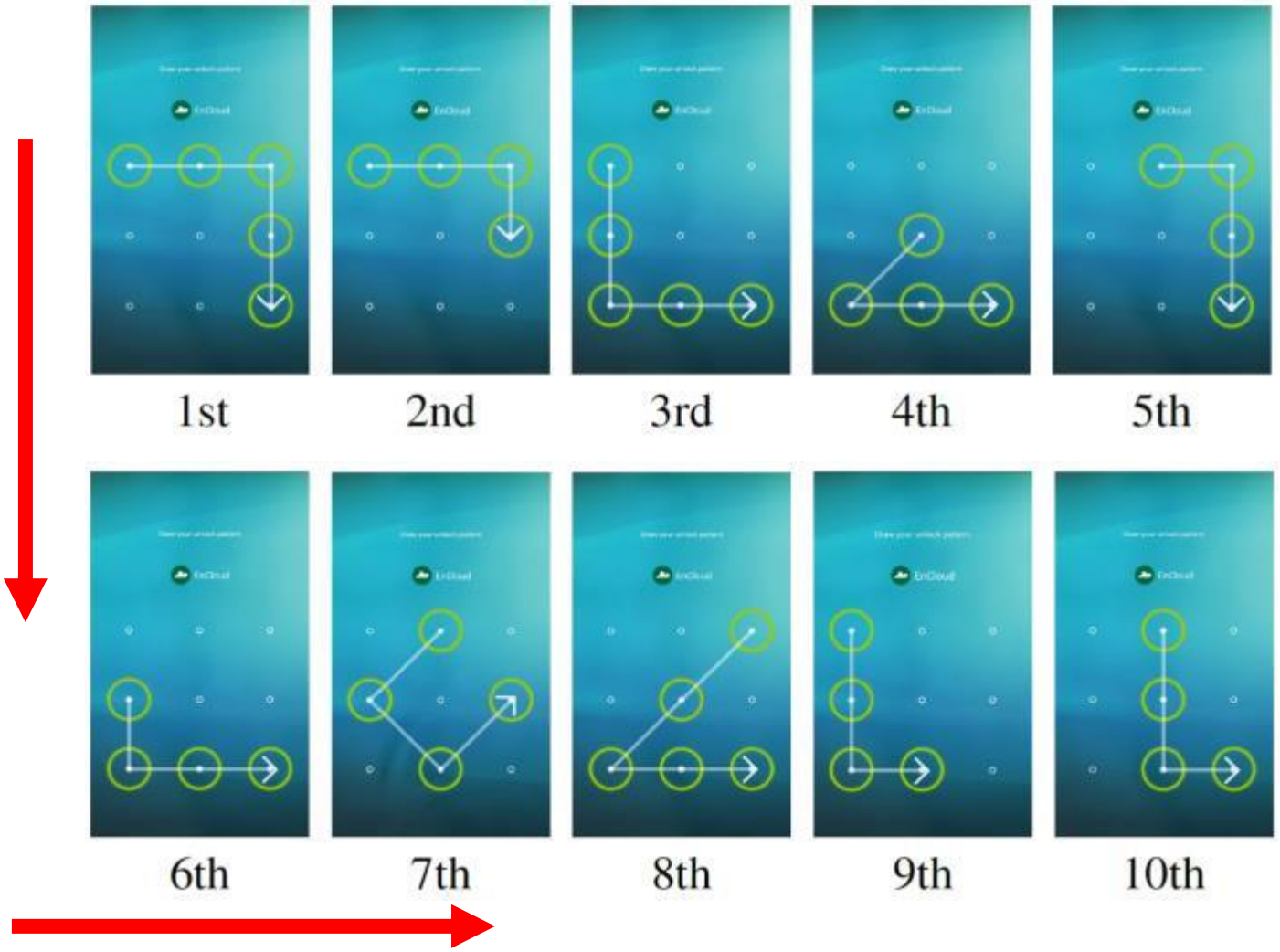
Short Length



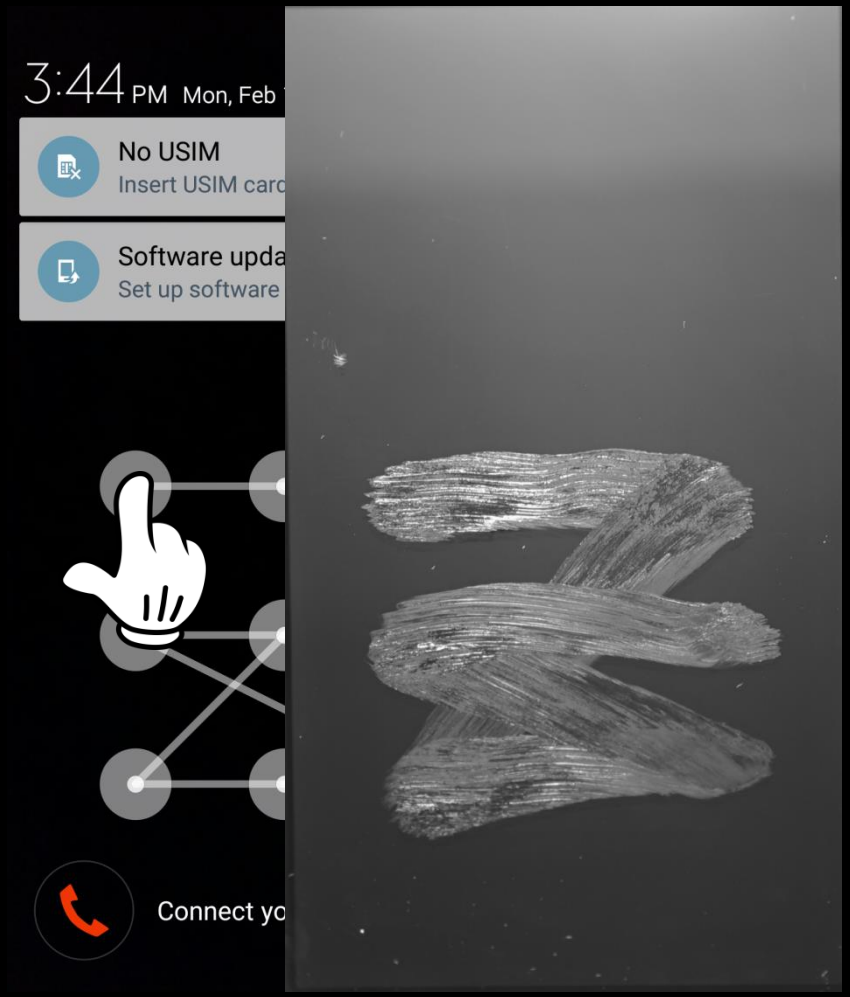
Small Number of Turns



Popular Directions

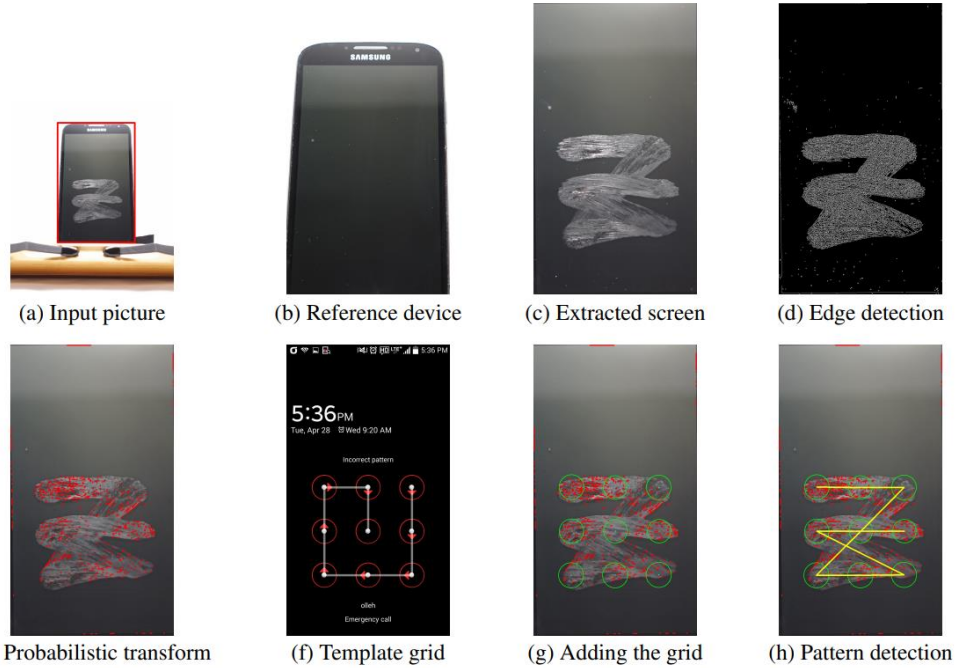


Side-Channel Attacks



Pattern Guessing with Smudge

We developed an automated pattern guessing attack by combining Markov model-based guessing attacks with computer vision-based smudge attacks to find a pattern secret with its smudge.



"Boosting the Guessing Attack Performance on Android Lock Patterns with Smudge Attacks," **ASIACCS 2017**

Pattern Guessing with Smudge

Our attack is effective. We can successfully recover a secret pattern with 74.17%. Even for the case after using Facebook, we can still recover a secret pattern with 31.94%.

	Unlocking only	Calling	Texting	Facebook
Avg. # of guessing attempts	4,634.66	6,811.83	9,783.01	13,130.74
Avg. # of guessing attempts (<= 20)	3.79	4.43	5.36	4.82
Total # of cracked patterns (<= 20)	**267 (74.17%)**	**189 (52.50%)**	**134 (37.22%)**	**115 (31.94%)**

"Boosting the Guessing Attack Performance on Android Lock Patterns with Smudge Attacks," **ASIACCS 2017**

Improving the Security of Android Pattern Lock (1/2)

- Changing a grid layout.
  - The use of a circle or random grid layout [1].
  - The use of a bigger grid (e.g., 4×4) layout [2].

User-chosen patterns were still **guessable**.

[1] Uellenbeck et al.,"Quantifying the security of graphical passwords: the case of android unlock patterns," **ACM CCS, 2013**.

[2] Aviv et al., "Is Bigger Better? Comparing User-Generated Passwords on 3x3 vs. 4x4 Grid Sizes for Android's Pattern Unlock," **ACM ACSAC, 2015**.

Improving the Security of Android Pattern Lock (2/2)

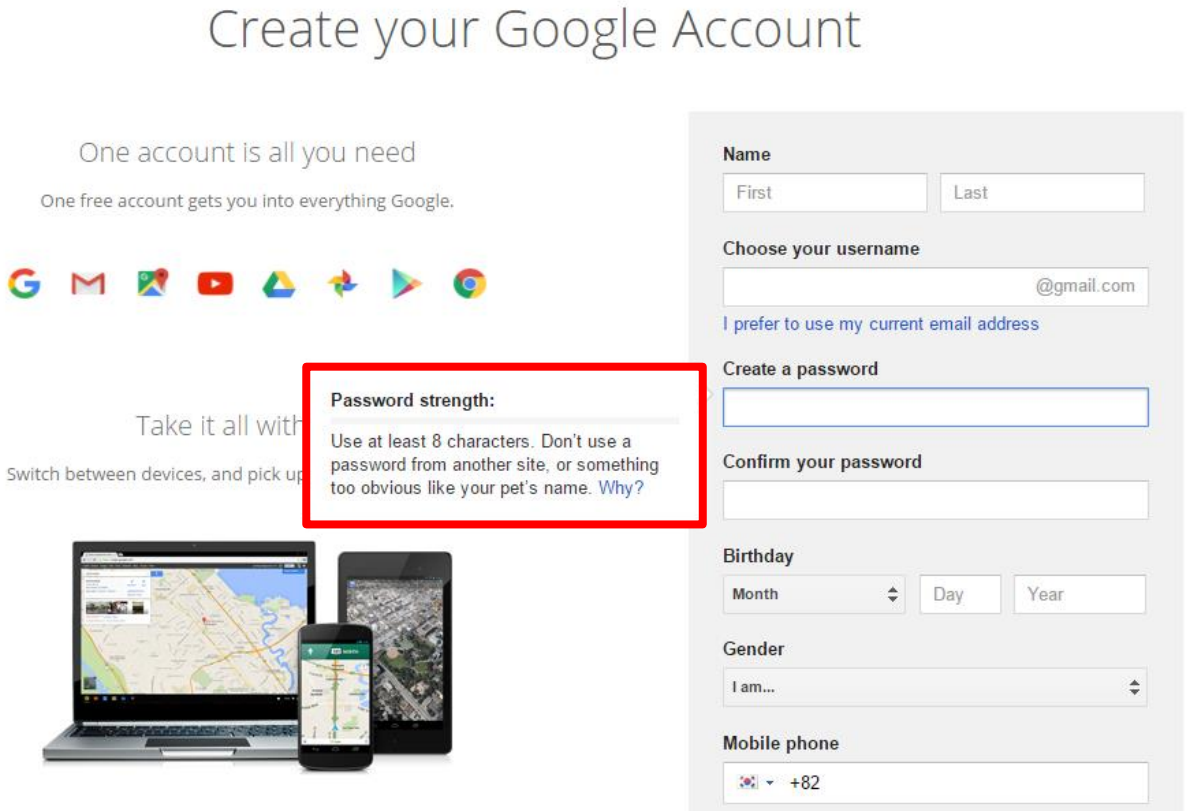
- The use of a strength meter.
  - The guessing entropy can be larger with meter support.

How can we avoid this?

"On the Effectiveness of Pattern Lock Strength Meters: Measuring the Strength of Real World Pattern Locks," **ACM CHI, 2015.**

Policies for Patterns

- Password selection policies can be used to avoid weak passwords.



Research Question

Our work was motivated by the following research question:

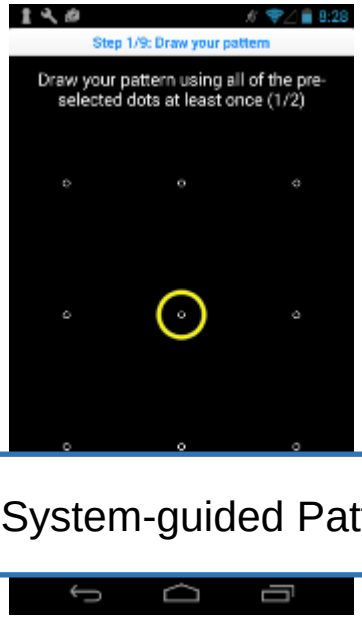
“Can we design effective security policies for Android patterns to improve their security without significantly compromising their usability?”

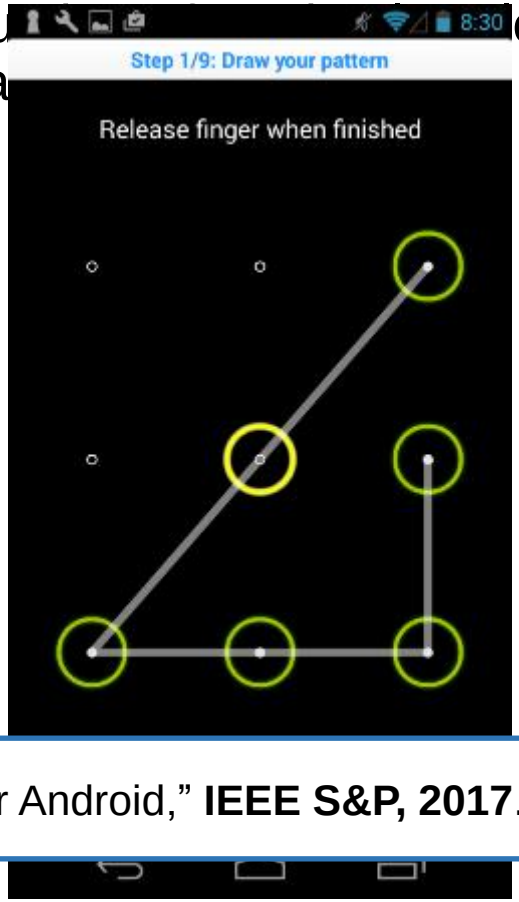
Design Principles

1. Minimize additional memorability burden on users.
2. Keep the authentication time similar to the original Android pattern locks.
3. Make it easy to learn and use.
4. Avoid significant software or hardware changes.

Solution: SysPal

- **SysPal** is a novel system-guided pattern lock scheme.
  - It mandates the use of a small nu[illegible]ected points (or lines) to avoid weak pa[illegible]

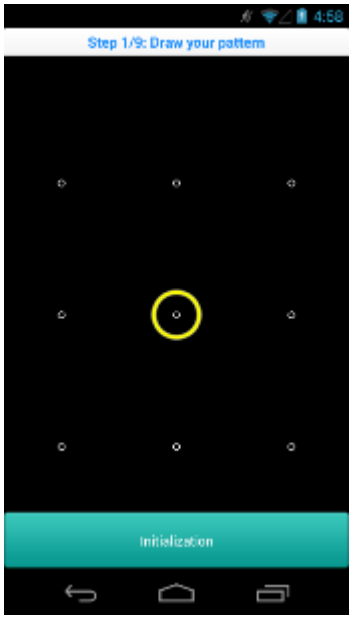




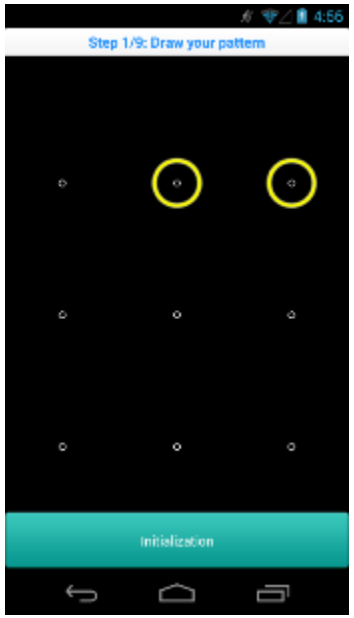
"SysPal: System-guided Pattern Locks for Android," **IEEE S&P, 2017**.

Hypothesis 1

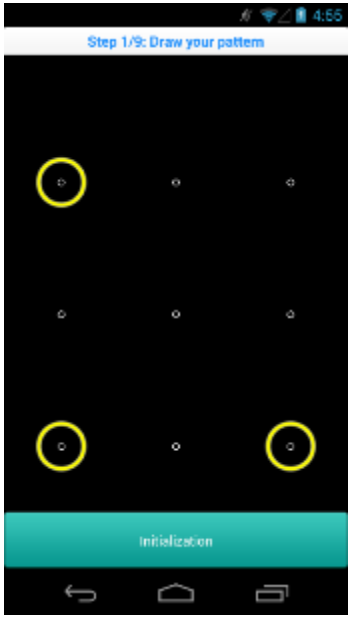
- The security of SysPal patterns strengthens with the increase in the number of mandated points.



One mandated point



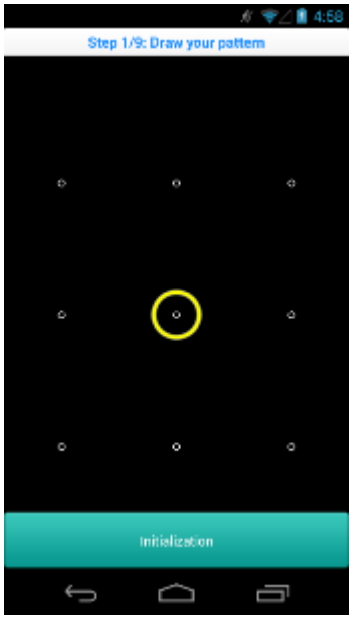
Two mandated points



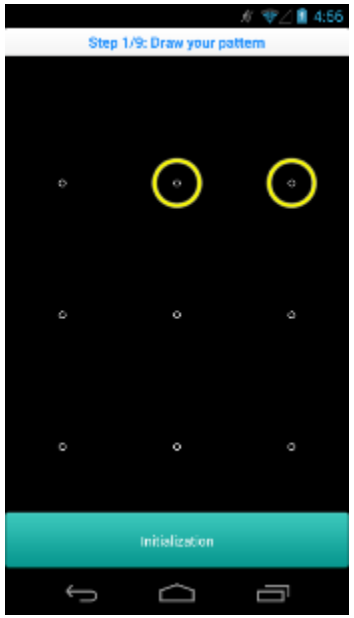
Three mandated points

Hypothesis 2

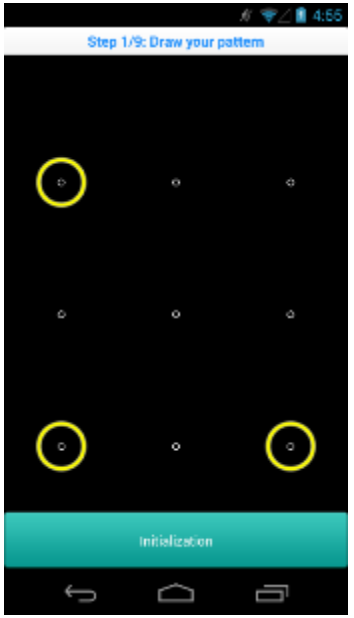
- The memorability of SysPal patterns decreases with the increase in the number of mandated points.



One mandated point



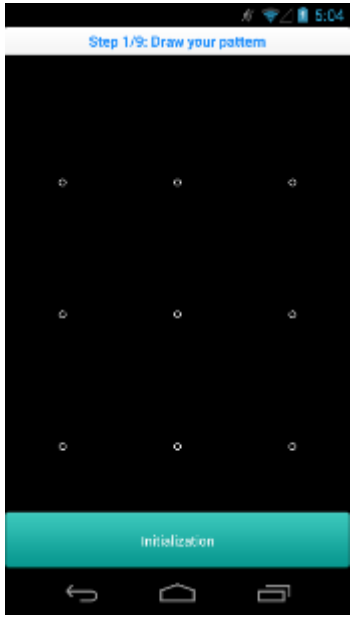
Two mandated points



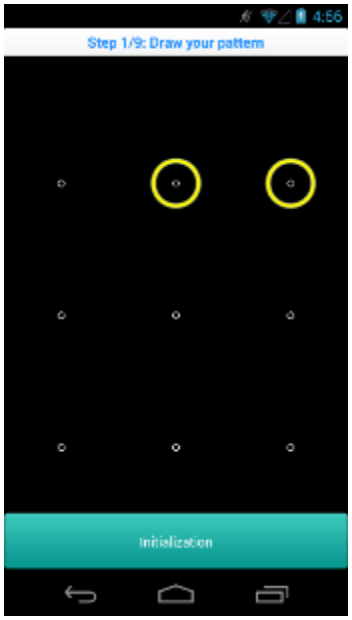
Three mandated points

Hypothesis 3

- A SysPal policy has statistically similar memorability to original Android patterns and *better security*.



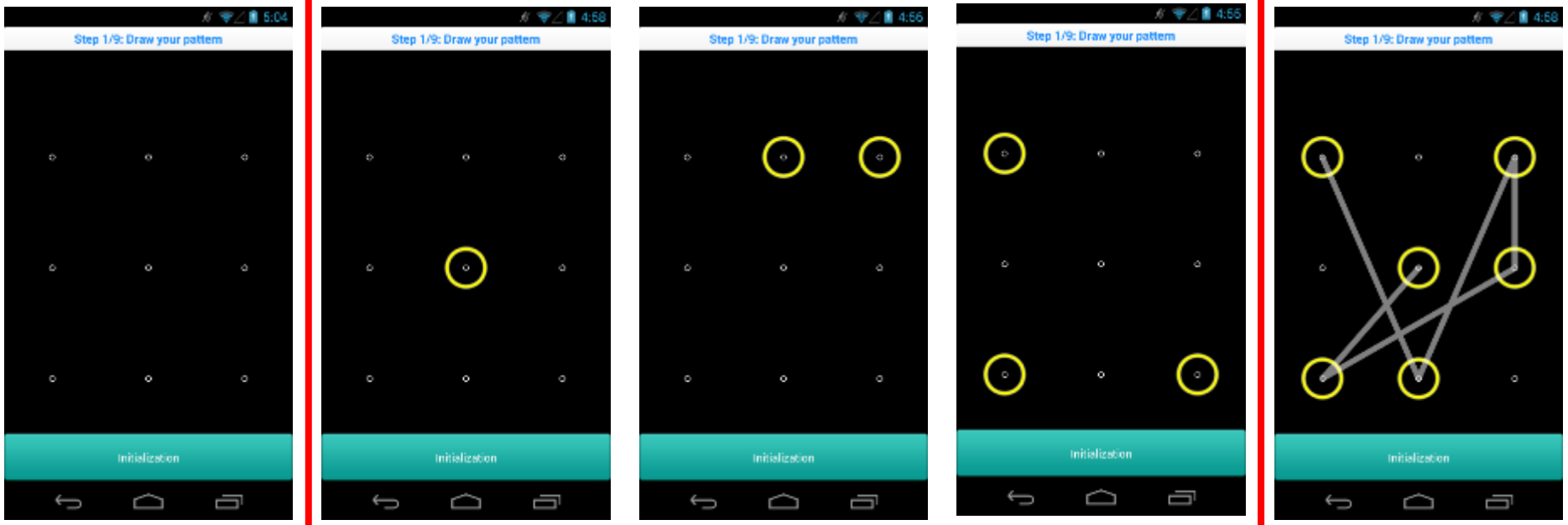
Original Android



Two mandated points

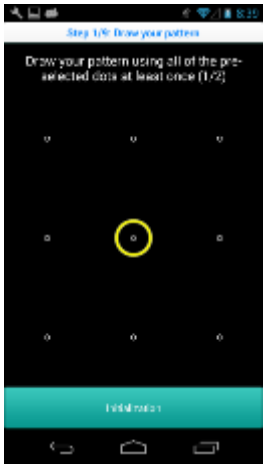
Five Policies

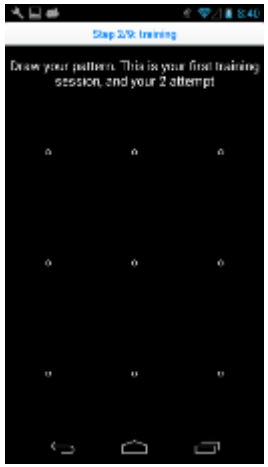
- The number of randomly-selected mandated point(s) must be used once upon selecting a pattern.



Study Design: Pattern Setup

- First large-scale study was conducted using Amazon Mechanical Turk, recruiting **1,717** participants.
- We developed an Android application to evaluate the SysPal policies in a realistic setting.
- Participants were asked to select a pattern under one of the five policies (randomly assigned).





Study Design: Recall Tests

- Complete two remembrance training tasks.
- Complete a graphical puzzle to wipe out their short-term memory [1].
- Conduct three recall tests after 5 minutes, 15 minutes, and 24 hours, respectively.
  - In each recall test, if a participant correctly draw his/her pattern within five chances, we regard that the participant passed the recall test.

[1] Atkinson et al., **"Human memory: A proposed system and its control processes,"** The psychology of learning and motivation, vol. 2, 1968.

Recall Success Rate

	Original	*1-Point*	*2-Point*	*3-Point*	*Random*
1st Test	382/384	326/331	340/342	320/326	276/334
	99.48%	**98.49%**	99.42%	98.16%	82.63%
2nd Test	365/384	317/331	330/342	312/326	265/334
	95.05%	95.77%	**96.49%**	95.71%	79.34%
3rd Test	278/384	232/331	252/342	231/326	169/334
	72.40%	70.09%	**73.68%**	70.86%	50.60%

Recall success rate for ***2-Point* is greater** than *Original* in the 2nd and 3rd test.
(corrected FET)

(# remaining participants) / (# initial participants)

Authentication Time

Policy	1st Test		2nd Test		3rd Test	
	μ	σ	μ	σ	μ	σ
Original	4.60	3.56	4.73	3.64	6.31	5.13
1-Point	4.26	2.76	4.07	2.76	6.53	6.75
2-Point	4.17	2.94	4.38	3.95	5.32	5.32
3-Point	4.47	4.30	4.52	4.77	5.79	6.13
Random	12.90	10.70	9.15	7.59	**13.65**	12.77

There is **no statistically significant difference** between *Original* and all SysPal policies. (corrected two-tailed unpaired t-test)

μ: mean, σ: standard deviation

Number of Attempts Made

Policy	1st Test		2nd Test		3rd Test	
	μ	σ	μ	σ	μ	σ
Original	1.14	0.50	1.11	0.36	1.24	0.74
1-Point	1.18	0.61	1.11	0.28	1.22	0.74
2-Point	1.12	0.47	1.10	0.40	1.29	0.83
3-Point	1.19	0.65	1.08	0.34	1.20	0.64
Random	2.26	1.52	1.70	1.15	**2.16**	1.57

Random participants drew their pattern twice on average.
(corrected two-tailed unpaired t-test)

μ: mean, σ: standard deviation

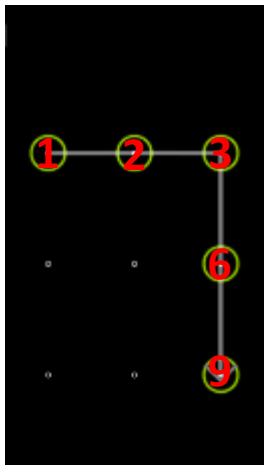
Measuring the Probability of Pattern Occurrence

- We analyzed the guessability of the collected patterns.
- We collected total 1,717 patterns. However, it is not sufficiently large to calculate guessability of all possible patterns.
- We estimated the probability of all possible pattern occurrences using an N-gram Markov model.

N-gram Markov Model

- How can we estimate the probability of a given pattern "1,2,3,6,9 ?"
- We can use an *N*-gram Markov model to estimate the probability of a pattern.

$$P(x_1, \ldots, x_m) = P(x_1, \ldots, x_{n-1}) \cdot \prod_{i=n}^{m} P(x_i | x_{i-n+1}, \ldots, x_{i-1})$$



- For example, using the 3-gram Markov model, we can calculate *P*(1,2,3,6,9) as

$$P(1,2,3,6,9) = P(2|\$, 1) \cdot P(3|1,2) \cdot P(6|2,3) \cdot P(9|3,6) \cdot P(@|6,9)$$

$$\frac{139}{440} \quad \frac{128}{169} \quad \frac{124}{170} \quad \frac{90}{148} \quad \frac{49}{131}$$

$$\approx 0.039697552$$

Guessing Entropy

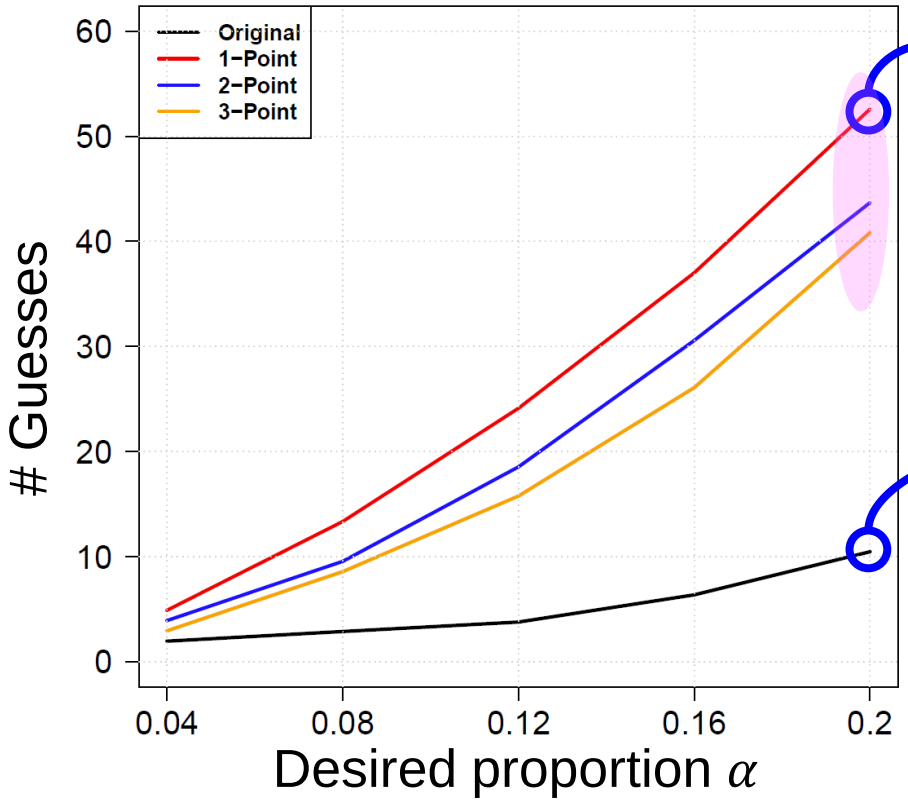
- We then compute the partial guessing entropy [1] for the distribution of all possible patterns.
- The patterns generated with the SysPal policies have significantly higher guessing entropy estimate than the original Android patterns.

Policy	α									
	0.1	*0.2*	*0.3*	*0.4*	*0.5*	*0.6*	*0.7*	*0.8*	*0.9*	*1.0*
Original	5.04	5.82	6.54	7.19	7.86	8.50	9.20	9.97	11.00	12.71
1-Point	7.54	8.19	8.67	9.16	9.67	10.21	10.82	11.57	12.44	13.67
2-Point	7.16	7.91	8.40	8.92	9.47	10.02	10.65	11.39	12.30	13.62
3-Point	6.95	7.81	8.52	9.12	9.69	10.29	10.96	11.71	12.59	13.79
Random	11.20	11.84	12.44	13.02	13.58	14.11	14.60	15.04	15.44	15.81
Random Patterns (U_{389112})	18.57	18.57	18.57	18.57	18.57	18.57	18.57	18.57	18.57	18.57
Real-world 4-digit PINs [17]	5.19	7.04	8.37	9.38	10.08	10.63	11.08	11.44	11.70	11.83
Random 4-digit PINs (U_{10000})	13.29	13.29	13.29	13.29	13.29	13.29	13.29	13.29	13.29	13.29
Real-world 6-digit PINs	10.71	13.32	14.03	14.50	14.92	15.36	15.86	16.49	17.14	17.53
Random 6-digit PINs ($U_{1000000}$)	19.93	19.93	19.93	19.93	19.93	19.93	19.93	19.93	19.93	19.93

[1] Bonneau, J. The science of guessing: analyzing an anonymized corpus of 70 million passwords. In Proceedings of Security and Privacy (SP), IEEE (2012).

Security of Android Patterns

- **<u>20% of the original Android patterns</u>** can be successfully guessed within 10 guessing attempts.



53 guessing attempts
(1-Point policy)

10 guessing attempts
(Original policy)

Pattern Cracking

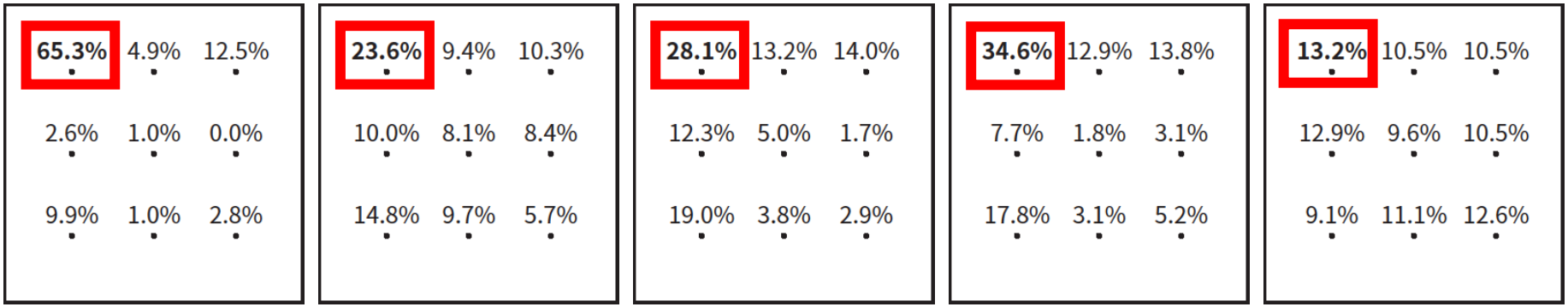
- We developed a dictionary of the top 20 frequently used patterns based on the *3*-gram Markov model.
- 32.55% of the o*riginal* Android patterns were successfully cracked within 20 guessing attempts.

	Original	*1-Point*	*2-Point*	*3-Point*	*Random*
Mean # of guessing attempts	5,492.97	3,803.01	2,993.18	3,740.18	47,445.51
Mean # of guessing attempts (≤ 20)	6.31	10.44	7.29	11.74	0.00
Mean % of cracked patterns (≤ 20)	**32.55%**	9.97%	9.36%	14.11%	0.00%

Statistically significant difference
(all $p < 0.001$, corrected FET)

Frequencies of the Points used as the Starting Point

- 65.3% of *Original* patterns started from the upper leftmost point.
- All SysPal policies also started from upper leftmost point.



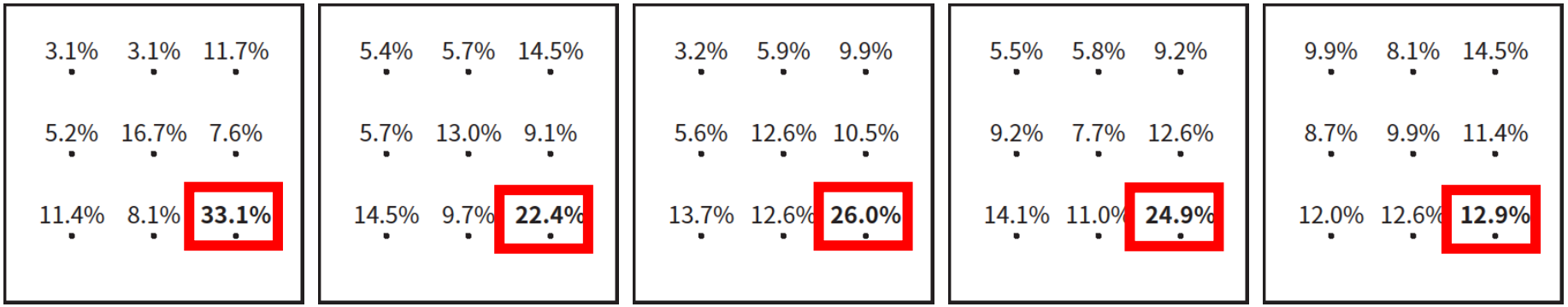
Original *1-Point* *2-Point* *3-Point* *Random*

Statistically significant difference

(all $p < 0.05$, corrected FET)

Frequencies of the Points used as the Ending Point

- Ending point is biased toward the lower rightmost point among all policies.

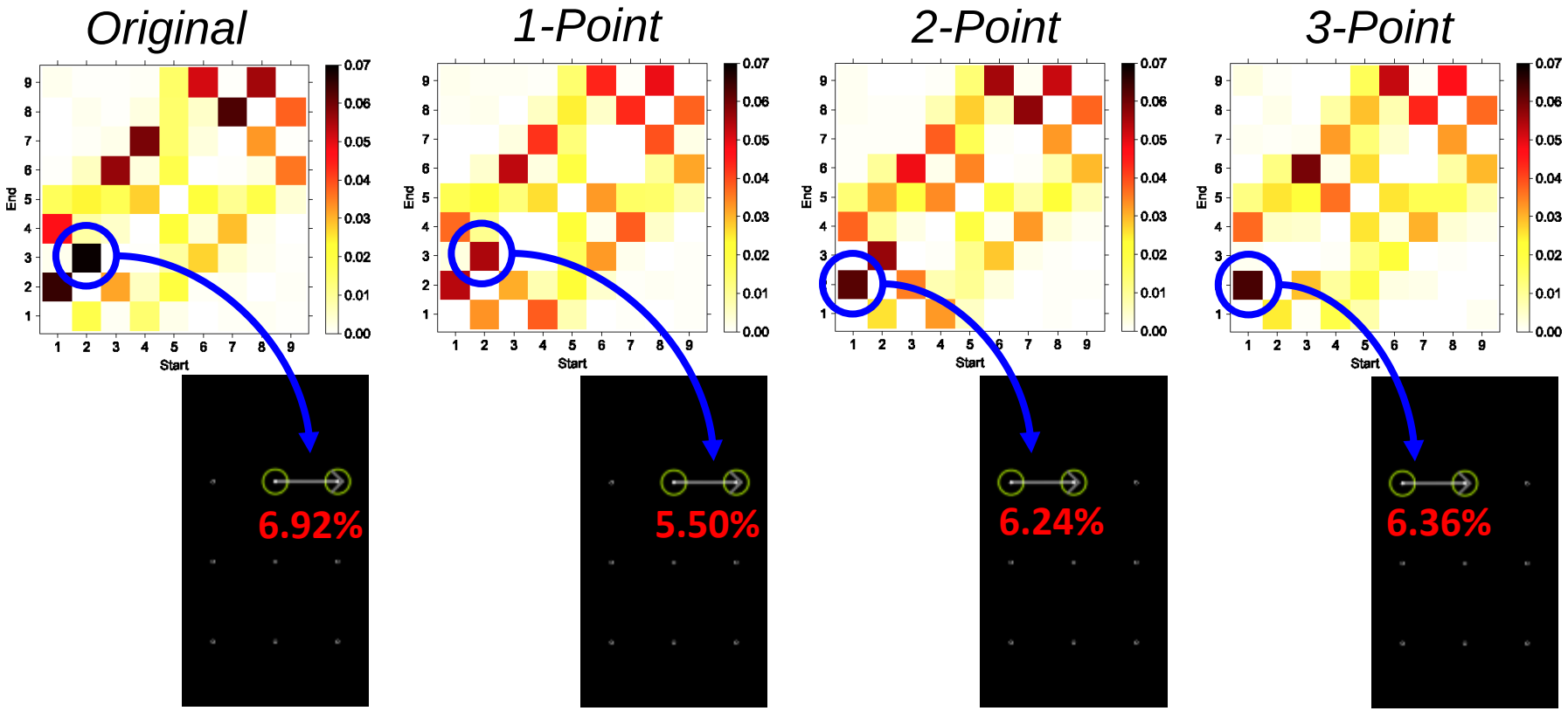


No statistically significant difference between all SysPal policies and Original.

(all p = 1.0, corrected FET)

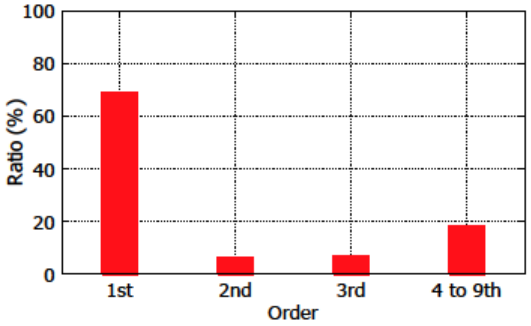
Frequencies of the Segments used

- *1-Point* seems more evenly distributed than *Original*.

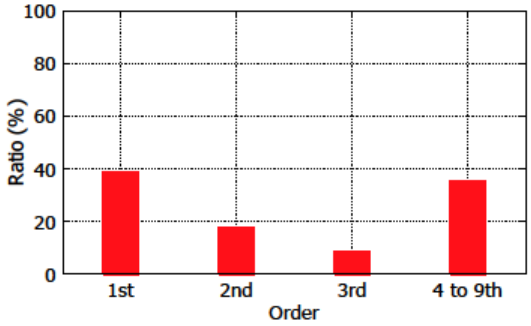


Proportion of the Mandated Points used

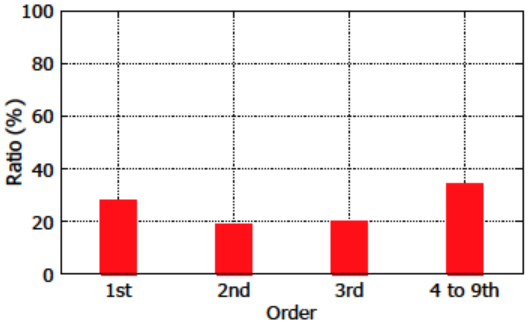
- A majority of SysPal participants used a given mandated point as the starting point of their patterns.
  - **68.9%** for *1-Point*, **38.5%** for *2-Point*, **17.4%** for *3-Point*.
- *2-Point* participants used 4th position frequently.
  - One of the mandated points could be used frequently as an ending point.



1-Point



2-Point

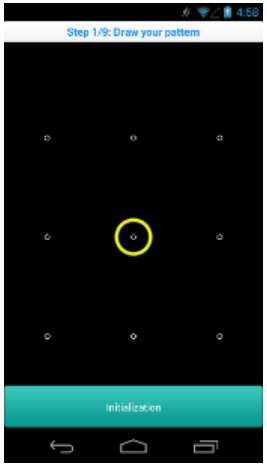


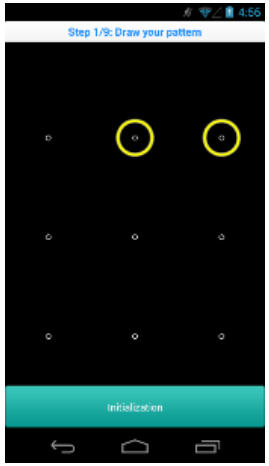
3-Point

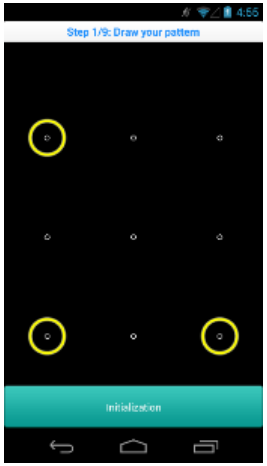
Security Improvements

H_1: The security of SysPal patterns strengthens with the increase in the number of mandated points

- The use of 3 mandated points is not helpful to improve the security of patterns.





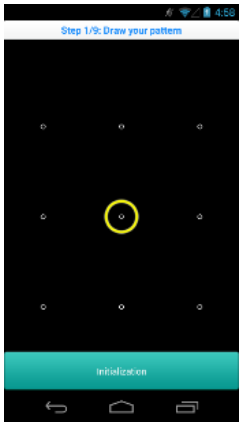


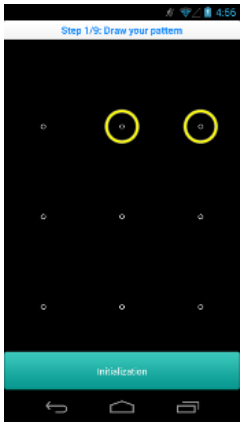
Guessing Entropy ($\alpha = 0.2$)	**8.19**	7.91	7.81
% of cracked patterns	9.97%	**9.36%**	14.11%

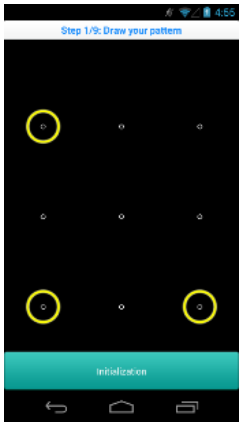
Recall Success Rate and Memorability Effects

H_2: The memorability of SysPal patterns decreases with the increase in the number of mandated points

- The effects of increasing the number of mandated points is *unclear.*







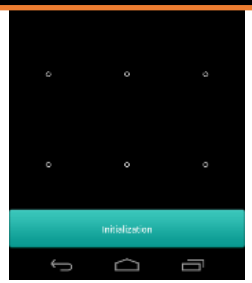
Recall Success Rate in the 3rd test	70.09%	**73.68%**	70.86%
Memorability	92.3%	**93.8%**	

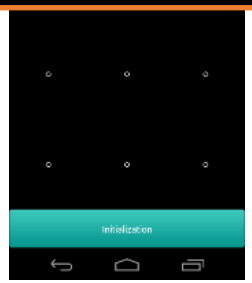
Replacing the Original Android Policy

H_3: A SysPal policy that shows no statistically significant difference in memorability against the original Android

SysPal policies can potentially **replace the current Android policy** without compromising too much usability

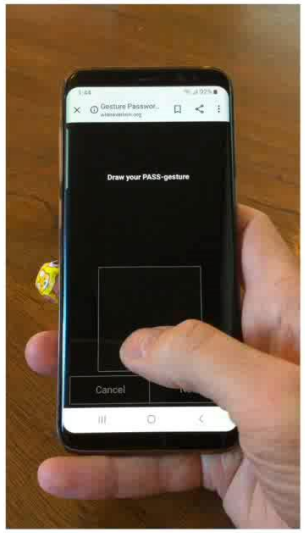
Security: All SysPal policies outperformed the *Original* policy.





Memorability	93.3%	**93.8%**
% of cracked patterns	32.55%	**9.36%**

Free-Form Gesture Password



Unmatched Pass-Gesture

Matched Pass-Gesture

Many users select **easy to guess** PINs and patterns to **lock their smartphones**.
Phone lock via **free-form pass-gestures** entered on the touchscreen may be **more secure**.

"Gesture Authentication for Smartphones: Evaluation of Gesture Password Selection Policies," **IEEE S&P, 2020.**

Problems with Gestures

- Recognition is not explicit.
- Recognition should be determined based on a similarity measure.

Reference

Matched

Not Matched

Deployed on Amazon MTurk

2594 valid participants

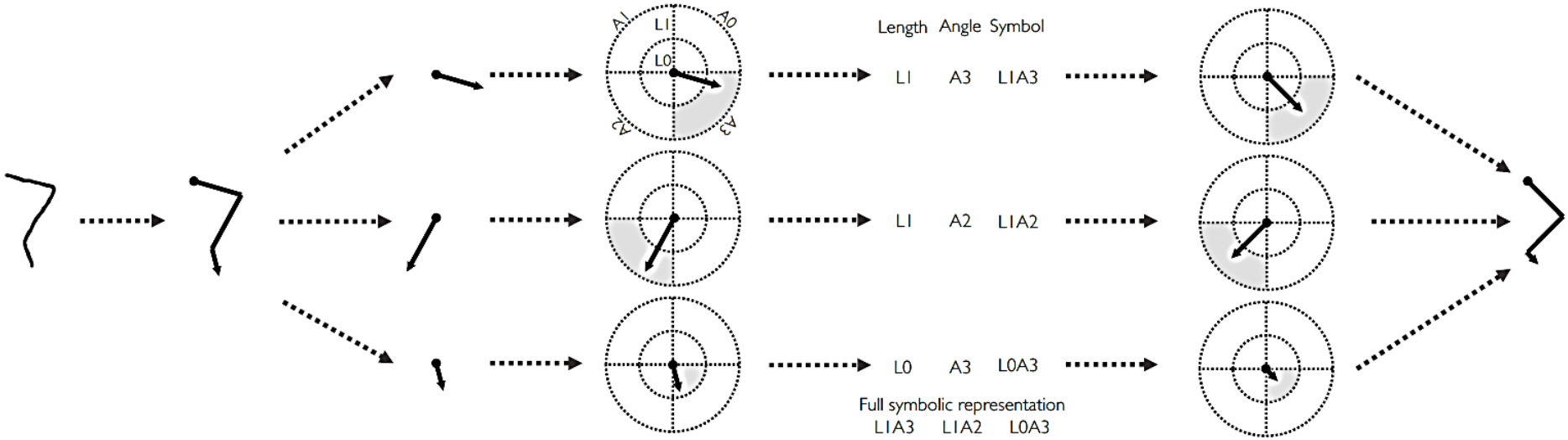
~13000 final gestures collected

~20000 gestures logged

5-25 times larger than prior datasets

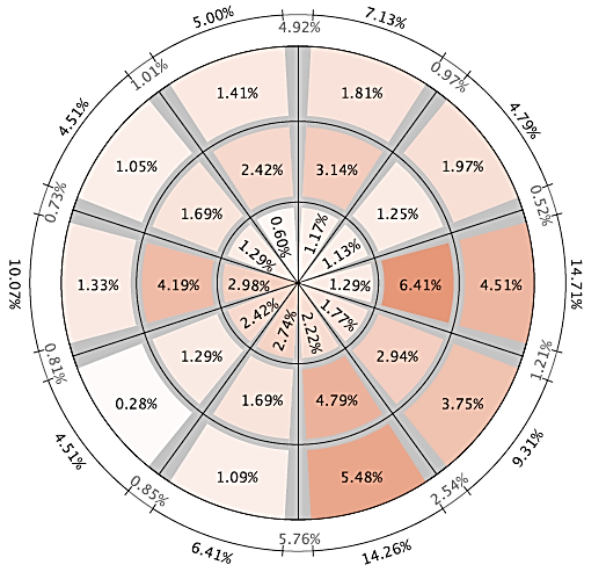
Gesture Discretization

- In theory, the gesture password space is not finite.
- To compute entropy, we need to discretize the password space of gesture passwords.

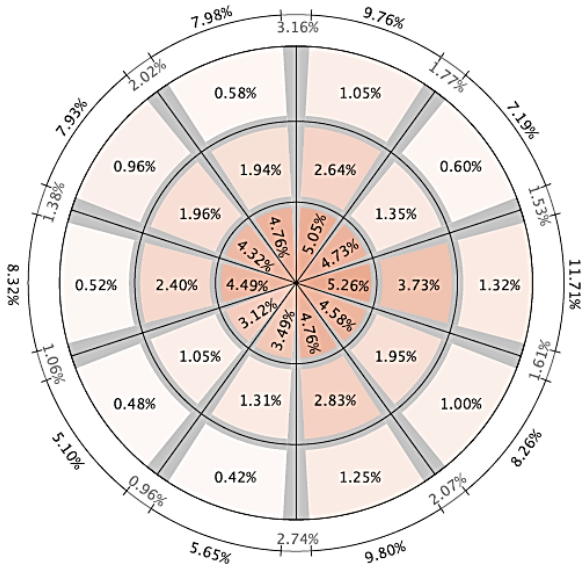


Finding the Optimal Model

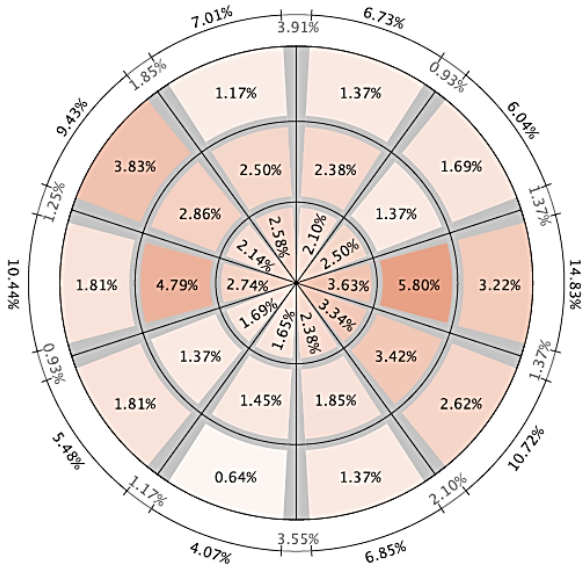
- We constructed 570 different n-gram Markov models.
- We optimized angle and length classes.
- The optimal model has 10 angles, 3 lengths and large overlapped boundary.



Initial Stroke



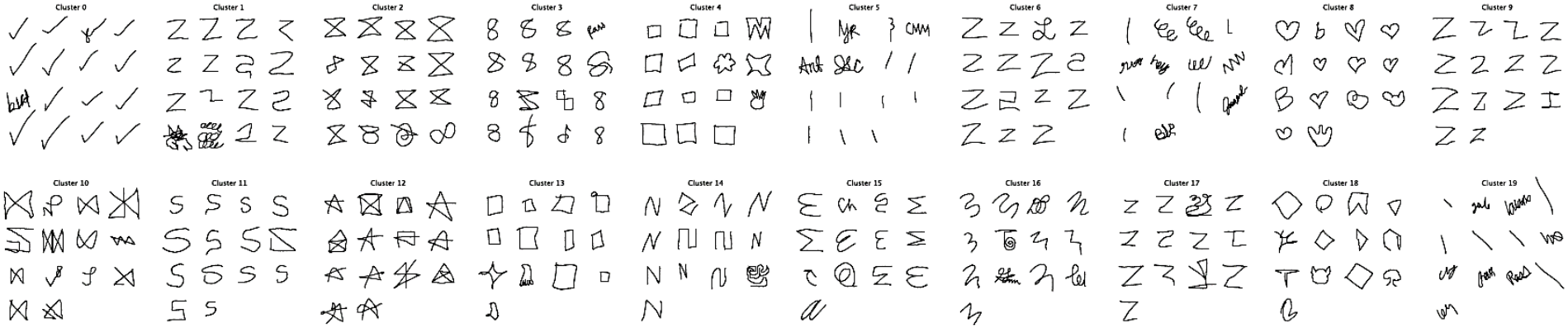
Center Strokes



Final Stroke

Clustering-based Attack

- We developed a novel attack using gesture clustering results.
  1. Compute similarity scores between all gestures.
  2. Cluster those gestures into groups based on the computed similarity scores.
  3. Find the top 20 largest clusters.
  4. Create the gesture password dictionary by selecting the representative one from each of the top 20 largest cluster.



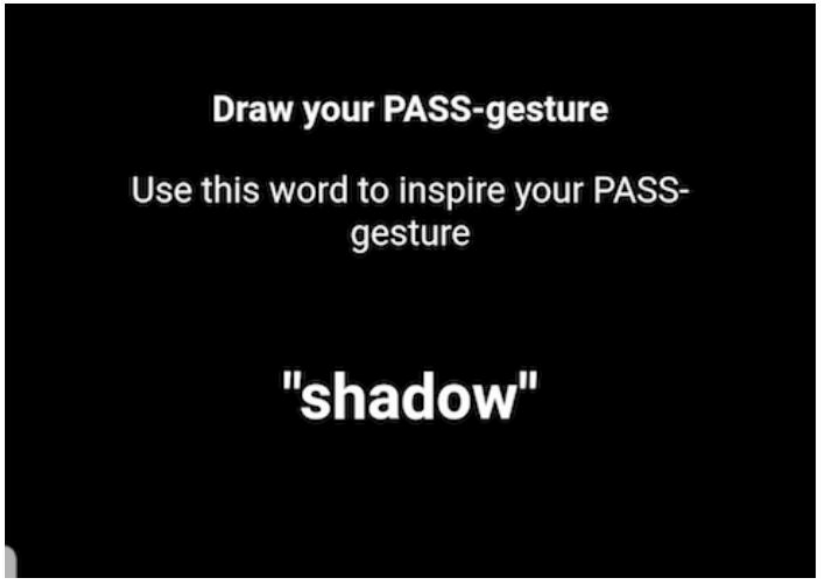
Dictionary of Gestures

We can crack about 22% of gesture passwords within 20 guesses.

Gesture Password Policies

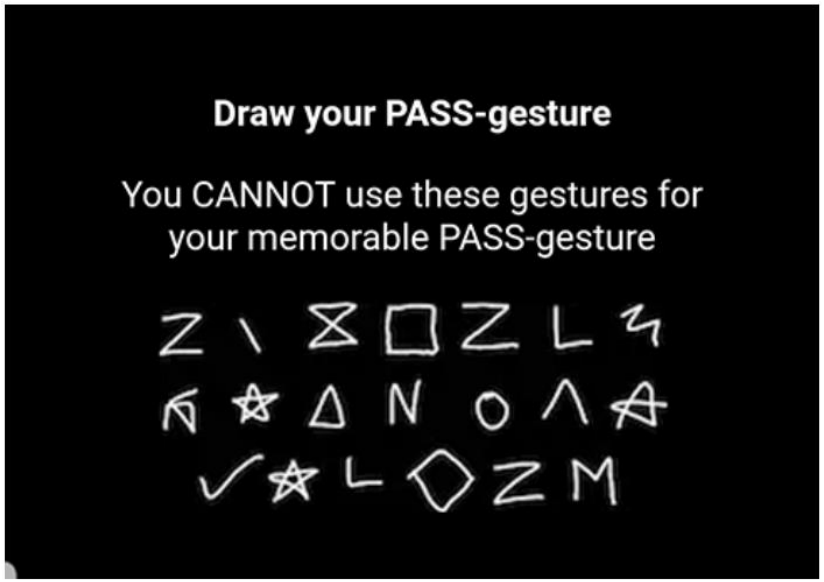
Lexical Policy

Words to inspire gestures



Blacklist Policy

Block common gestures



Password cracking results

- Baseline policy: 23% cracking rate
- Lexical policy: 34% cracking rate
- Blacklist policy: 17% cracking rate
- Lexical + Blacklist (consolidated) policy: 15% cracking rate

Our **blacklist** and **consolidated** policies would improve the security of gesture passwords, while our lexical policy reduces it.

Conclusions

- User-chosen graphical passwords can be vulnerable to password guessing attacks.
- Addition of randomness can be helpful to improve the security of graphical password schemes without compromising the usability of those schemes.
- Compliance might be better than recommendation to prevent poor security practices.

Any questions?

Hyoungshick Kim

hyoung@skku.edu