REPLICATE: Countering Concurrent Login Attacks in “Just Tap” Push-based Authentication

A Redesign and Usability Evaluation

Jay Prakash
Clarice Chua, Tanvi Thombre, Andrei Bytes, Mohammed Jubur, Nitesh Saxena, Lucienne Blessing, Jianying Zhou, and Tony Q.S Quek
This paper highlights issues and proposes a solution for usable push notification based 2FA.

1. The current 2FAs, especially Just Tap to authenticate, are not secure enough.

2. Proposes usable methods to fix the Vulnerabilities.

Our method: REPLICATE to Authenticate.
Current solutions do not have balance between Usability and Security.
Why?
Problem: State of TOTP 2FA

TOTPs apps are vulnerable to screen overlays and Accessibility based attacks

In-app generator requires effort, takes time to auth and in vulnerable to remote login attempts.

- Malware with accessibility permissions can capture credentials entered by the user on mobile banking apps, read or generate SMS messages, and even read Two-Factor Authentication (2FA) codes generated by authenticator apps!
Problem: State of TOTP 2FA

TOTPs apps are vulnerable to screen overlays and Accessibility based attacks

Malware Apps:
- Cerberus
- TrickBot,
- DEFENSOR ID,
- TeaBot,
- Oscorp,
- Toddler.

Problem: State of TOTP 2FA

TOTPs apps have high friction in use and adoption

- User experience is affected due to time, distraction and errors while performing copying and typing of the OTP codes.
Push based 2FAs are vulnerable to concurrency and overlays attacks. 

If an attacker & user logs in at the same time, the token device receives two pushes.

User tends to approve either or attacker’s push (sophisticated attack) [1, Asia CCS 2021]

1. Mohammed Jubur, Prakash Shrestha, Nitesh Saxena, Jay Prakash, Bypassing Push-based Second Factor and Passwordless Authentication with Human-Indistinguishable Notifications, ASIA CCS 2021
Problem: State of Push based 2FA

Push to Approve Notifications are not differentiable
Problem: State of Push based 2FA

Push to Approve Notifications are not differentiable
Solution?

Key Idea:

Remove static and fixed responses to Push

Random Interaction at login prompt

Replicate using token device
Our Solution: REPLICATE to Prove and Auth

Key Idea: REPLICATE

Show randomized interaction at the screen
Ask user to respond
Reduced Concurrency attack
REPLICATE: System Architecture

User application/browser → Application Server → PushAuth Service → Token

Login → Auth → Send Push → Sign Push → < Activity Id, IMUs > → Authenticate/Rejected → Accept/Deny → Randomised Activity

Allow/Deny → Show
REPLICATE: Forms for Study

Show randomized interaction at the screen

Key Drag

Application Server

PushAuth Service

Token Device
Show randomized interaction at the screen

Move a Shape to Auth (a) Login window display and (b) Phone authentication push screen overlay.
Show randomized interaction at the screen

**Randomized Keypad**

Randomized Keypad to Auth (a) Login window display and (b) Phone authentication push screen overlay
REPLICATE: Forms for Study

Show randomized interaction at the screen

Choose a Colored Button

Choose a Colored Button to Auth (a) Login window display and (b) Phone authentication push screen overlay
REPLICATE: Forms for Study

Show randomized interaction at the screen

Tap on Black Button

Tap on Black Button (a) Login window display and (b) Phone authentication push screen overlay
REPLICATE: Forms for Study

Show randomized interaction at the screen

Draw Pattern

Draw Unlock Pattern (a) Login window display and (b) Phone authentication push screen overlay
Figma was used as a base for the remote study.
REPLICATE: Study Design

Pre-Test Survey

• What is your age group? a) 20 and below b) 21-30 c) 31-40 d) 41-50 e) 51 and above.

• On a scale of 0 to 4, how would you rate your level of familiarity in using your laptop? (0 being the least familiar and 4 being the most familiar).

• On a scale of 0 to 4, how would you rate your level of familiarity in using your phone? (0 being the least familiar and 4 being the most familiar.)

• Do you know what second-factor authentication is? a) Yes or b) No.

• In which of the following areas have you used a two-factor authentication? You can select more than one option. a) I have not done a two-factor authentication before, b) Banking, c) Email, d) Social Media, and e) Others.

• Have you come across a push-based TFA? Some examples are shown in the image below. a) Yes b) No
REPLICATE: Study Design

**Quantitative**
- All positive SUS Survey

**Qualitative**
- User Journey Mapping
REPLICATE: Study Design

**Quantitative**
- All positive SUS Survey

**Qualitative**
- User Journey Mapping (UJM)

Trial 2 and 3, UJM and overall post test interview.
Qualitative Study: Generating UJM

We proposed Hybrid Emotrack: Inspired from Plutchik Emotion Wheel and Emotrak

Plutchik Emotion Wheel [1]

Emotrak by UEGroup [1]

Qualitative Study: Generating UJM

We proposed Hybrid Emotrack: Inspired from Plutchik Emotion Wheel and Emotrak

Qualitative Study: Generating UJM

We proposed Hybrid Emotrack: Inspired from Plutchik Emotion Wheel and Emotrak Hybrid Emotrack with emojis as hint for emotions
REPLICATE: Study Design

1. Laptop as login device & Phone as a token
2. Both login and authentication on a phone

Just Tap to Authenticate
Quantitative Analysis

**Task Success**

<table>
<thead>
<tr>
<th>Authentication Method</th>
<th>Task Success Rate (in %)</th>
<th>95% Confidence Interval (Adjusted Wald Method)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Just Tap</td>
<td>100</td>
<td>(90,100)</td>
<td>✔️</td>
</tr>
<tr>
<td>Choose Number</td>
<td>100</td>
<td>(90,100)</td>
<td>✔️</td>
</tr>
<tr>
<td>Randomized Keypad</td>
<td>93</td>
<td>(76,99)</td>
<td></td>
</tr>
<tr>
<td>Key Drag</td>
<td>100</td>
<td>(90,100)</td>
<td>✔️</td>
</tr>
<tr>
<td>Shapes</td>
<td>100</td>
<td>(90,100)</td>
<td>✔️</td>
</tr>
<tr>
<td>Colored Button</td>
<td>97</td>
<td>(82,100)</td>
<td></td>
</tr>
<tr>
<td>Black Button</td>
<td>100</td>
<td>(90,100)</td>
<td>✔️</td>
</tr>
<tr>
<td>Draw Unlock Pattern</td>
<td>100</td>
<td>(89,100)</td>
<td></td>
</tr>
</tbody>
</table>

**TABLE 1. Task Success Rates with 95% CI (L2P)**

- If participant managed to reach “Unlocked”/“Authenticated” in one try without external helps gets 1 score, 0 otherwise.
Quantitative Analysis

Task Time

The time taken (in seconds) for the participant to authenticate successfully; task time = end time - start time.
Just Tap and Black button method seem to be competitive.

<table>
<thead>
<tr>
<th>Authentication Method</th>
<th>Q1</th>
<th>Mean</th>
<th>Median</th>
<th>Q3</th>
<th>95% Confidence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Just Tap</td>
<td>4.19</td>
<td>4.69</td>
<td>4.65</td>
<td>5.12</td>
<td>(4.37,5.01)</td>
</tr>
<tr>
<td>Choose Number</td>
<td>4.24</td>
<td>4.87</td>
<td>4.78</td>
<td>5.08</td>
<td>(4.50, 5.24)</td>
</tr>
<tr>
<td>Randomized Keypad</td>
<td>5.60</td>
<td>6.02</td>
<td>5.98</td>
<td>6.44</td>
<td>(5.69, 6.35)</td>
</tr>
<tr>
<td>Key Drag</td>
<td>4.82</td>
<td>5.66</td>
<td>5.35</td>
<td>5.86</td>
<td>(5.07, 6.25)</td>
</tr>
<tr>
<td>Shapes</td>
<td>4.67</td>
<td>5.73</td>
<td>5.66</td>
<td>6.27</td>
<td>(5.25, 6.21)</td>
</tr>
<tr>
<td>Colored Button</td>
<td>4.60</td>
<td>5.61</td>
<td>5.27</td>
<td>5.89</td>
<td>(5.09, 6.13)</td>
</tr>
<tr>
<td>Black Button</td>
<td>4.31</td>
<td>5.36</td>
<td>4.59</td>
<td>5.36</td>
<td>(4.61, 6.11)</td>
</tr>
<tr>
<td>Draw Unlock Pattern</td>
<td>4.35</td>
<td>5.41</td>
<td>5.06</td>
<td>5.99</td>
<td>(4.82, 6.00)</td>
</tr>
</tbody>
</table>

**TABLE 2. STATISTICS OF AUTHENTICATION TIME (IN SECONDS)**
Quantitative Analysis

Task Time

- User got familiarized with the trials and time plateaued.
Quantitative Analysis

Efficiency

- Efficiency: Task success/ task time.
- Black Button and Draw Unlock Pattern’s task efficiency were comparable to existing solutions.
• Tap on Black Button fairs similar to Just Tap method and better than the rest
### Quantitative Analysis

#### Combined Analysis

<table>
<thead>
<tr>
<th>Authentication Method</th>
<th>Task Success</th>
<th>Time</th>
<th>Task Efficiency</th>
<th>SUS Score</th>
<th>Favourite Count</th>
<th>Less Favourite Count</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Just Tap</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>Choose Number</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>14</td>
</tr>
<tr>
<td>Randomised Keypad</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>3</td>
<td>4</td>
<td>8</td>
<td>39</td>
</tr>
<tr>
<td>Key Drag</td>
<td>1</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>26</td>
</tr>
<tr>
<td>Shapes</td>
<td>1</td>
<td>7</td>
<td>6</td>
<td>8</td>
<td>7</td>
<td>7</td>
<td>36</td>
</tr>
<tr>
<td>Coloured Button</td>
<td>7</td>
<td>5</td>
<td>7</td>
<td>6</td>
<td>3</td>
<td>6</td>
<td>34</td>
</tr>
<tr>
<td>Black Button</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>Draw Pattern</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>7</td>
<td>2</td>
<td>24</td>
</tr>
</tbody>
</table>

- Based on ranking 1 to 8.
- Tap on Black Button fairs similar to Just Tap method and better than the rest.
Quantitative Analysis

Compared to PIN 2FA

- REPLICATE fairs better than PIN based 2FAs across all dimensions including time and SUS scores.
Qualitative Analysis

Ease of Execution and Cognitive Effort

- Black button method faired much well compared to others in the usability ranking.

  P3: “I prefer the Black Button method. For PIN-based method, I had to look at the screen at least 3 times to ensure I was doing it right. For Black button, I didn’t have to refer back.”

  P14: “It is easy enough for you to just press 1 time. The rest requires dragging and more actions. This is the fastest.”

  P11: “This is the easiest to understand. By far the best for understanding and carrying out.”

  P30: “The buttons were all grey, and only the one I had to press was in black. The difference in the coloring was fun and helped me easily identify which button to press. Also, after I pressed the correct button, then the exact color in the instructions showed on my phone, which helped me to recognize that I did it correctly.”
Qualitative Analysis

Perception of Security

• 13/40 cited lack of security in Just Tap to authenticate method.

  P9: “It is easy but secure. Just Tap is easy but not secure. For Black Button you only have 1/9 chance to randomly click on it. Easiest and more safe.”

• Participants appreciated forcing to think method of REPLICATE.

  P34: “Simple but makes you think twice. Thinking twice is important because I want to be aware of what I am doing.”

  P25: “I like that it is fairly simple but relatively much more secure than current simple TFA methods.”
Qualitative Analysis

Inclusivity

- Colored Button noticed concerns over color blindness.

  **P12:** “In consideration of a minority of people that are colorblind, this method won’t be feasible.”

- Drawing patterns witnesses issues about unfamiliarity.

  **P37:** “Troublesome, especially for old people. This is okay for unlocking phone because we are used to the same pattern, but in this case, it is always a different pattern.”
Participants felt encouraged to try out 2FAs as PIN based methods are boring.

P33: “I like this. Usual OTP methods are very boring. This method is like a small puzzle. Nice shapes. More fun and less boring, I like this.”

Participants prefer the more engaging or fun method if everything else is comparable. Key Drag, Shapes and Colored Button were deemed fun by some participants.

P13: “Once in a while, it will be a very fun way. Looking at the key was more fun. The shapes look simple, but the key has a nicer display. It is a unique way to unlock because it literally has a key to unlock.”
Qualitative Analysis

User Journey Map

• Tap on Black Button fairs similar to Just Tap method and better than the rest
Comparisons with other 2FAs

<table>
<thead>
<tr>
<th>Schemes</th>
<th>Usability</th>
<th>Deployability</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Memory-wise-Effortless</td>
<td>Scalable-for-Users</td>
<td>Resilient-to-Physical-Observation</td>
</tr>
<tr>
<td>PIN</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Just Tap</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>REPLICATE</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>


• REPLICATE fairs better than PIN based 2FAs across all dimensions including time and SUS scores.
Key Drag Demo
Successful Authentication Attempt
Countering Concurrent Login Attacks in “Just Tap” Push-based Authentication: A Redesign and Usability Evaluations

Jay Prakash1,2, Clarice Chun Qing Yu1,2, Tanvi Ravindra Thombr2, Andrei Bytev1, Mohammed Jahan3, Nitesh Saxena1, Lucienne Blessing2, Junying Zhou2, and Tony Q-S Quek2

1Silence Laboratories, Singapore
2Singapore University of Technology and Design, Singapore
3University of Alabama at Birmingham, USA

Abstract—In this paper, we highlight a fundamental vulnerability associated with the widely adopted “Just Tap” push-based authentication in the face of a concurrency attack, and propose the method REPLICATE, a redesign to counter this vulnerability. In the concurrency attack, the attacker launches the login session at the same time the user initiates a session, and the user may be hooded, with high likelihood, into accepting the push notification which corresponds to the attacker’s session, thinking it is their own. The attack stems from the fact that the login notification is not explicitly mapped to the login session running on the browser in the Just Tap approach. REPLICATE attempts to address this fundamental flaw by having the user approve the login attempt by replicating the information presented on the browser session over to the login notification, such as by moving a key in a particular direction, choosing a particular shape, etc. We report on the design and a systematic usability study of REPLICATE. Even without being aware of the vulnerability, in general, participants placed multiple variants of REPLICATE in competition to the Just Tap and fairly above PIN-based authentication.

1. Introduction

Push notification based authentication, such as seen in solutions like, Duo-Push [1] or Authy [2], has witnessed a sharp rise in adoption in the past few years. It has been deployed as second-factor authentication (TFA) or password-less authentication. A device is first enrolled as a token device and associated with an account (service) pair. Next, whenever a user attempts to log in to an application or web-service, and enter the correct credentials, the token device receives a push notification. When the user opens/reads the notification, a session overlay requests if the user wants to approve or deny the login attempt (Figure 1). The usability pain point is well relieved by this “Just Tap” push-based authentication compared to traditional one-time PIN (OTP) based TFA as there is no need to copy the PIN code from the device to the login terminal/browser. Hence, being more usable than OTP-based TFA, push notification assisted authentication has witnessed growing user adoption as reflected in the success of Duo Security and commercial adoption by software and service giants like Twitter, Yahoo, Google [1], [3], [4] and academic entities.

However, Just Tap push-based authentication has a fundamental and easy-to-exploit vulnerability, which we call the “concurrency attack.” In this attack, the malicious actor launches the login session at the same time as the user. The user may then be hooded, with high likelihood, into approving the push notification corresponding to the attacker’s session, allowing the attacker to successfully access the user’s account. This will break the second-factor security offered by Just Tap TFA, assuming the attacker has already compromised the first factor, the password (e.g. via hacked password databases). In the case of Just Tap password-less authentication, the attacker does not need to compromise the user’s password. In the concurrency attack, the attacker’s goal is to confuse the legitimate user with two or more similar push notifications. As presented in Figure 3, the push notification prompts ask the user to tap on the “Yes” or “No” button. The only differentiating information from login attempt of an attacker is the location name, usually given as coordinates. However, this information is too coarse and gives ample scope of the attack. Also, the attacker can spoof the location. The legitimate user would have no definite way to identify the correct push notification or even the possibility of an adversarial login, and they will most likely approve the attacker’s notification.

To better understand this attack context, we simulated the situation of concurrent logins with 75 pairs of legitimate users and the attacker. The study set consisted of diverse personas, including undergraduate students of different streams, faculty, corporate employees from both business and information technology (IT) domains, and retired and old users. Statistically, only 5% of people (mostly comprising of IT employees and a few university students) raised doubts when receiving such notifications in the concurrency attack. Most of the people approved