

# Exploring the Security and Privacy Impacts of Using 2FA Apps

*Conor Gilsenan*



Electrical Engineering and Computer Sciences  
University of California, Berkeley

Technical Report No. UCB/EECS-2025-49

<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2025/EECS-2025-49.html>

May 13, 2025

Copyright © 2025, by the author(s).  
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

Exploring the Security and Privacy Impacts of Using 2FA Apps

By

Conor Gilsenan

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Doctor Serge Egelman, Co-Chair  
Professor David Wagner, Co-Chair  
Associate Professor Raluca Ada Popa

Spring 2025

# Exploring the Security and Privacy Impacts of Using 2FA Apps

Copyright 2025

By

Conor Gilsenan

## Abstract

Exploring the Security and Privacy Impacts of Using 2FA Apps

By

Conor Gilsenan

Doctor of Philosophy in Computer Science

University of California, Berkeley

Doctor Serge Egelman, Co-Chair

Professor David Wagner, Co-chair

The Time-based One-Time Password (TOTP) algorithm is a two-factor authentication (2FA) method that is widely deployed, but forces people to face a critical usability challenge: maintain access to the secrets stored within the TOTP app, or risk getting locked out of their accounts. Prior work has regularly confirmed that TOTP users are concerned about account lockout and, therefore, has called for improvements to backup and recovery mechanisms. However, the existing backup and recovery options for TOTP users were not well explored in the literature, which is a necessary starting point from which to design improvements. My work fills this gap and explores the functionality of existing backup mechanisms for TOTP users and how they get used in the real world.

We reverse engineered the top 22 general purpose Android TOTP apps and found that many backup implementations allowed the developer or other third-parties to access personal user information, had serious cryptographic flaws, and/or allowed the app developers to access the TOTP secrets in plaintext. Most backup strategies also ended up placing trust in the same technologies that TOTP 2FA is meant to supersede: passwords, SMS, and email. Next, we surveyed 330 current and former users of popular TOTP apps. A significant portion lacked basic awareness of account lockout risks. Two-thirds of current users had cloud backups enabled, exposing them to some of the security and privacy issues previously uncovered. Many of them did not know the feature existed nor that it was enabled, raising questions about whether they provided informed consent. The majority of current users were uncomfortable with anyone being able to read data from their cloud backups. About one-third had experienced account lockout, but most regained access quickly using alternative 2FA mechanisms (e.g., SMS 2FA). Notably, 13% of current users had no TOTP backup plan at all, putting 10+ million people at heightened risk of account lockout when extrapolated across the 100s of millions of TOTP app users.

To Jordan—without your unwavering support, this would neither have started nor finished.  
Thank you.

# Contents

<b>Contents</b>	<b>ii</b>
<b>List of Figures</b>	<b>iv</b>
<b>List of Tables</b>	<b>vi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Research Overview . . . . .	3
<b>2 Background on TOTP</b>	<b>4</b>
<b>3 Related Work</b>	<b>7</b>
3.1 2FA Adoption . . . . .	7
3.2 2FA and Account Recovery . . . . .	10
<b>4 Security and Privacy Analysis of Backup Mechanisms in TOTP Apps</b>	<b>16</b>
4.1 Methods . . . . .	17
4.2 Results . . . . .	22
4.3 Discussion . . . . .	35
4.4 Supplementary Materials . . . . .	39
<b>5 Exploring Account Lockout Among TOTP Users</b>	<b>40</b>
5.1 Methods . . . . .	41
5.2 Results . . . . .	48
5.3 Limitations and future work . . . . .	68
<b>6 Conclusion</b>	<b>70</b>
6.1 In practice, TOTP does not escape the vulnerabilities of SMS 2FA . . . . .	70
6.2 TOTP 2FA should be retired in favor of FIDO2 . . . . .	73
6.3 TOTP apps should be improved during the migration to FIDO2 . . . . .	73
<b>Bibliography</b>	<b>77</b>

<b>A</b>	<b>Supplemental Materials for Chapter 4</b>	<b>88</b>
A.1	Full App Names and Versions . . . . .	88
A.2	Base32 decryption heuristic . . . . .	90
A.3	Google Play Search Terms . . . . .	90
A.4	Network Traffic Snippets . . . . .	91
<b>B</b>	<b>Supplemental Materials for Chapter 5</b>	<b>104</b>
B.1	2FA descriptions used in screening survey . . . . .	104
B.2	Contents of Main Survey . . . . .	105
B.3	Participant instructions to check whether cloud backups are enabled or disabled	161



# List of Figures

1.1	Screenshot of the Google Authenticator TOTP app. . . . .	2
2.1	Standard workflow for a user to enable TOTP on their account for the first time. . . . .	4
2.2	Contents of a TOTP setup QR-code [60]. . . . .	5
4.1	Plaintext warning in the <i>Aegis Authenticator</i> app. . . . .	23
5.1	Prompt asking about which 2FA options a participant had used and the description for the TOTP option. . . . .	42
5.2	Example screenshot shown to participants when they were asked which TOTP apps they had used. . . . .	43
5.3	Example of account recovery codes often displayed when a user is enabling TOTP on their account. . . . .	46
5.4	Example of a setup QR-code displayed when a user is enabling TOTP on their account. . . . .	47
5.5	Counts of which app current users ( $n = 293$ ) were most familiar with, segmented by platform (Android & iOS). . . . .	49
5.6	The account recovery strategies that Lockouts ( $n = 104$ ) used when trying to regain access to their account(s), grouped by people who used a single strategy and people who used multiple strategies. . . . .	55
5.7	Cloud backups guess and actual usage among current users ( $n = 293$ ). . . . .	57
5.8	Cloud backups usage among current users ( $n = 293$ ), segmented by TOTP app. Note: some users did not report enabled/disabled. . . . .	58
5.9	Where people stored recovery codes ( $n = 155$ ) and setup QR-codes ( $n = 36$ ), grouped by people who stored in only that location and people who also stored in additional locations. Participants could select multiple storage locations. . . . .	63
5.10	How many current users ( $n = 293$ ) believed the app developer and other third-parties could read TOTP fields from cloud backups created by their app. dnk = "I do not know" . . . . .	67
5.11	How comfortable current users ( $n = 293$ ) would be if TOTP data in cloud backups could be read by the app developer and other third-parties. . . . .	68
6.1	PayPal email promoting TOTP 2FA as more secure than SMS 2FA. . . . .	71

6.2	Twitter restricted SMS 2FA to paying customers only in March 2023. . . . .	72
-----	--	----

# List of Tables

3.1	Rates of 2FA adoption recorded by Conor Gilsean. Sources available at: <a href="https://allthingsauth.com/2fastats">allthingsauth.com/2fastats</a> . . . . .	8
4.1	Overview of the backup mechanisms supported in each app. <b>Y*</b> indicates that there is a serious security flaw in the implementation and/or usage of cryptography (see Section 4.2.3). <b>Y^</b> indicates support for multiple types of encrypted file exports (see Section 4.2.3.4). Values in parentheses were obtained from documentation and observation only (see Section 4.3.4). . . . .	21
4.2	Overview of the backup mechanisms that automatically sync data to the cloud. <b>Yes*</b> indicates a serious security flaw in the implementation and/or usage of cryptography (see Section 4.2.3). <b>Y^</b> indicates the field is conditionally included in the backup as plaintext (see Section 4.2.5). Values in parentheses were obtained from documentation and observation only (see Section 4.3.4). . . . .	25
4.3	Cryptographic details of app backup mechanisms. The asterisk (*) indicates that the app leaks the encryption key and/or password to the same service which stores the ciphertext, allowing that service to decrypt the TOTP backup (see Section 4.2.3.3). Square brackets indicate the min and max of a range, inclusive. Values in parentheses were obtained from documentation and observation only (see Section 4.3.4). The abbreviations for KDF configurations are: SHA/PKCS12/PBKDF2 (i = iterations), scrypt (N= CPU/memory cost, r = block size, p = parallelism), and Argon2 (m = memory, t = time/iterations, p = parallelism). . . . .	26
5.1	Demographics of main survey participants ( $n = 330$ ). . . . .	50
5.2	Top 10 recovery approaches that current users ( $n = 293$ ) said they would try. Descriptions could have multiple labels. Kupper-Hafner IRR = 0.899. . . . .	52
5.3	Two-tailed Fisher's exact tests showed significant differences in lockout rates among various groups ( $N = 330$ ). The p-values are listed below contingency tables (a) and (b). . . . .	54
5.4	Top 10 reasons the enabled cohort ( $n = 197$ ) enabled cloud backups. Each response could have multiple labels. Kupper-Hafner IRR = 0.926. . . . .	60
5.5	Top 10 reasons the disabled cohort ( $n = 84$ ) disabled (or never enabled) cloud backups. Each response could have multiple labels. Kupper-Hafner IRR = 0.938. . . . .	61

5.6	Top 10 recovery approaches that high-risk users ( $n = 37$ ) said they would try. Descriptions could have multiple labels. Kupper-Hafner IRR = 0.899. . . . .	65
A.1	Mapping of abbreviated app names used throughout the paper to identifying metadata and version information. . . . .	89

## Acknowledgments

If one proverb captures my PhD journey, it is the well-known saying: “It takes a village.”

First and foremost, I am grateful to my wife, Jordan Fischer. Without your unwavering support, this foolhardy adventure would quite literally neither have started nor finished. You have my sincerest and deepest thanks. On to our next adventure!

It is difficult to express in words how much I value the support of my dear colleagues and friends Nathan Malkin, Nikita Samarin, and Noura Alomar. Each of you has taught me so much and been there to help me push through the lows and celebrate the highs. Our friendship is unquestionably my most valuable achievement during my time at Berkeley.

My research would not have been possible without the collaboration of my coauthors, Fuzail Shakir and Diana Kohr. Fuzail, thank you for helping me reverse engineer all of those Android apps and review their cryptographic implementations. Your expertise, humor, and positive attitude were essential to completing the project. Diana, thank you for your relentless support of the user survey project. Your help refining questions, writing data analysis scripts, and coding participant responses was instrumental. I learned so much working together and thoroughly enjoyed every minute of our collaboration.

I extend my sincerest thanks to Coye Cheshire and everyone from DRTW. Coye, your passion for conducting good science is only surpassed by your love of teaching and connecting with students. If I ever become a professor, I will strive to emulate your humor, communication skills, and overall ability to inspire. Thank you for supporting my research, serving on my Quals committee, and leading DRTW for so many years. To the members of DRTW: thank you for reading early drafts of my work, providing thoughtful and constructive feedback, and helping illuminate paths forward when I was stuck, overwhelmed, and trying to do too much. DRTW has improved my research in countless ways and I am honored to have been an “honorary member” of the iSchool for so many years.

I also want to express my appreciation to those who taught me in earlier stages of my academic journey—without you, I would not be where I am today. David Farrell, thank you for fostering my earliest interests in computer programming and teaching me how to learn. Terry Harvey, thank you for teaching me so much as a freshman and for freely offering your support and advice over the years. Christopher Rasmussen, thank you for introducing me to research as an undergraduate.

I am extremely grateful to the University staff who helped me navigate the bureaucracy and provided critical support when I most needed it. Thank you: Carissa Caloud, Allison Torres, and Jean Nguyen.

Last, but certainly not least, thank you to all my friends and family who supported me throughout this adventure. I appreciate you occasionally asking how things were going because you were genuinely curious, letting the conversation drop when I gave the stereotypical reply of “same old, same old,” and engaging deeply when I chose to delve into the details of my research.

# Chapter 1

## Introduction

Though passwords are still the predominant authentication method [13, 52], it is well established that most people cannot create strong passwords [39, 41, 101]. Two-factor authentication (2FA)—logging in with a combination of at least two of something you know, something you physically have, and something you are—has consistently been shown to significantly increase the security of online accounts compared to the use of a password alone [106, 25]. Use of biometrics—something you are—is typically reserved for local, on-device authentication because transmitting biometrics over the network is extremely risky; they are inherent characteristics of a person and cannot be rotated if an attacker obtains a copy. Therefore, online accounts are often protected with 2FA implementations that combine the knowledge factor and the physical possession factor. There are many different methods of 2FA that utilize these factors and each has different tradeoffs in terms of security, privacy and usability. In practice, many online services often implement multiple 2FA mechanisms.

Over the years, researchers have tried to identify the most effective approaches to get users to proactively protect their online accounts by voluntarily enabling 2FA [89, 3, 4, 48, 20]. However, 2FA adoption rates have remained stubbornly low across the web [44]. For example, as recently as 2021, GitHub reported a 2FA adoption rate of 15.5% and Twitter just 2.3%. Duo Security reported that “a minority of respondents (32%) report using 2FA on all applications that offer it” [15]. Recently, more and more sites have shifted their approach from *asking* users to enable 2FA to *requiring* that they do so [70, 7]. Given this new wave of users enabling 2FA on their accounts, it remains critical to explore the issues that users face with prevalent methods of 2FA.

The Time-based One-Time Passwords (TOTP) algorithm [68] is one prevalent 2FA method that is widely deployed, including at some of the largest sites on the Internet [1], and used by hundreds of millions of people to protect their accounts. In addition to a username/password, TOTP requires the user to enter a one-time password (OTP) to login. In practice, these OTPs are typically 6-digit codes that are generated by a *TOTP authenticator app*<sup>1</sup> installed on the user’s phone/mobile device, which change to a unique value periodi-

---

<sup>1</sup>For brevity, I will refer to “TOTP apps” throughout this dissertation.

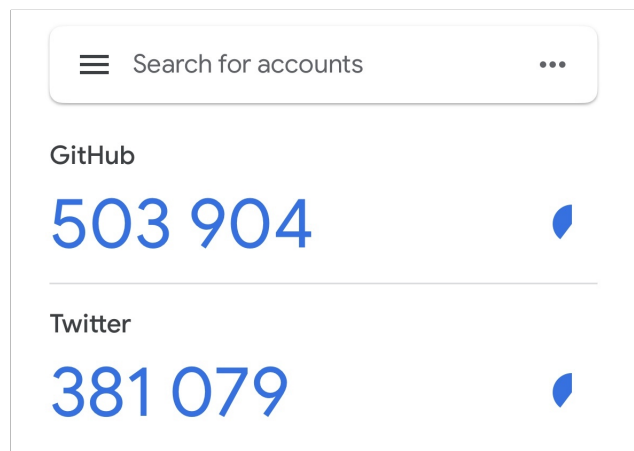


Figure 1.1: Screenshot of the Google Authenticator TOTP app.

cally (typically, every 30 seconds). Figure 1.1 shows a screenshot of Google Authenticator, the most popular TOTP app, with time-based OTPs displayed for two different example accounts.

TOTP is often promoted as a more secure method of 2FA than sending OTPs via SMS [26], which inherits many vulnerabilities from the telecommunication networks on which SMS operates [65, 64]. In contrast, TOTP apps can generate OTPs without any network connection by using a locally-stored secret obtained from the server during setup. While TOTP is widely supported by sites [1] and enabled by users [63], it does have several drawbacks. Like many other 2FA methods, including SMS 2FA, TOTP is vulnerable to phishing and social engineering attacks. Attackers who obtain a TOTP secret can generate valid OTPs for a user's account, so websites have the responsibility of storing them securely and striving to prevent data breaches. Many TOTP users have also reported issues with the setup process to enable TOTP on their accounts [92], usability challenges retrieving an OTP when authenticating to accounts [92], and concerns about what to do if they lost access to the TOTP app [21].

Despite these downsides, TOTP remains one of the most commonly deployed 2FA methods [1]. The usability challenges associated with enabling TOTP and the annoyances with daily usage could be addressed with thoughtful UX improvements by developers of websites and TOTP apps. However, the risks that arise from a user losing access to the TOTP app are much more fundamental problems inherent to the TOTP scheme. TOTP places a critical usability burden on users: maintain access to these shared secrets, or risk getting locked out of their account(s). In practice, these secrets are routinely lost; people lose their devices, buy new ones, or uninstall their TOTP apps. Without the TOTP shared secret, the user cannot generate the OTPs required to authenticate and could face account lockout.

To combat this usability nightmare, many TOTP apps provide custom backup mechanisms to help users recover from device loss. These backup and recovery features are critical

usability enhancements, but were understudied in the literature. It is necessary to understand the security and privacy impacts these features have on the millions of people who use these TOTP apps on a daily basis in order to design improvements. This dissertation fills that gap by exploring the functionality of existing backup mechanisms for TOTP users and how they get used in the real world.

## 1.1 Research Overview

This dissertation presents the findings from two comprehensive studies that explored the security and privacy impacts of using TOTP apps using a mixed-methods approach, including reverse engineering, measurements, user surveys, quantitative analysis, and qualitative analysis.

Chapter 2 provides important background on how TOTP 2FA works, which sets the foundation for the rest of the dissertation. There are many different 2FA mechanisms deployed in the real world along side, or in place of, TOTP and each has tradeoffs in terms of security, privacy and usability. Chapter 3 provides an overview of the authentication and recovery landscape in which TOTP exists.

Chapter 4 presents a study in which we defined an assessment methodology for conducting systematic security and privacy analyses of the backup and recovery functionality of TOTP apps. We reverse engineered all general purpose Android TOTP apps in the Google Play Store with at least 100k installs that implemented a backup mechanism ( $n = 22$ ). Our findings show that most backup strategies ended up placing trust in the same technologies that TOTP 2FA is meant to supersede: passwords, SMS, and email. Many backup implementations shared personal user information with third parties, had serious cryptographic flaws, and/or allowed the app developers to access the TOTP secrets in plaintext. We present our findings and recommend ways to improve the security and privacy of TOTP 2FA app backup mechanisms.

Chapter 5 presents findings from a large scale survey ( $n = 330$ ) of current and former users of popular TOTP apps. A significant portion lacked basic awareness of account lockout risks. Two-thirds of current users had cloud backups enabled, exposing them to some of the security and privacy findings detailed in Chapter 4. Many of them did not know the feature existed nor that it was enabled, raising questions about whether they provided informed consent. While many believed app developers *could* access their backup data, they were uncomfortable with this data collection in practice. About one-third had experienced account lockout, but most regained access quickly, frequently relying on alternative 2FA mechanisms (e.g., SMS 2FA) supported by the websites on which they had accounts. Notably, 13% of current users had no TOTP backup plan at all, putting 10+ million people at heightened risk of account lockout when extrapolated across the 100s of millions of TOTP app users.

Chapter 6 summarizes key findings about the security and privacy risks for users of TOTP apps. I argue that, in practice, TOTP 2FA does not provide much more security than SMS 2FA for most people and that websites should implement FIDO2 instead of TOTP.



## Chapter 2

# Background on TOTP

This chapter provides important background on how TOTP 2FA works, which will be referenced throughout this dissertation. Figure 2.1 outlines the 5 steps of the standard workflow for a user, Alice, to enable TOTP on their account for the first time.

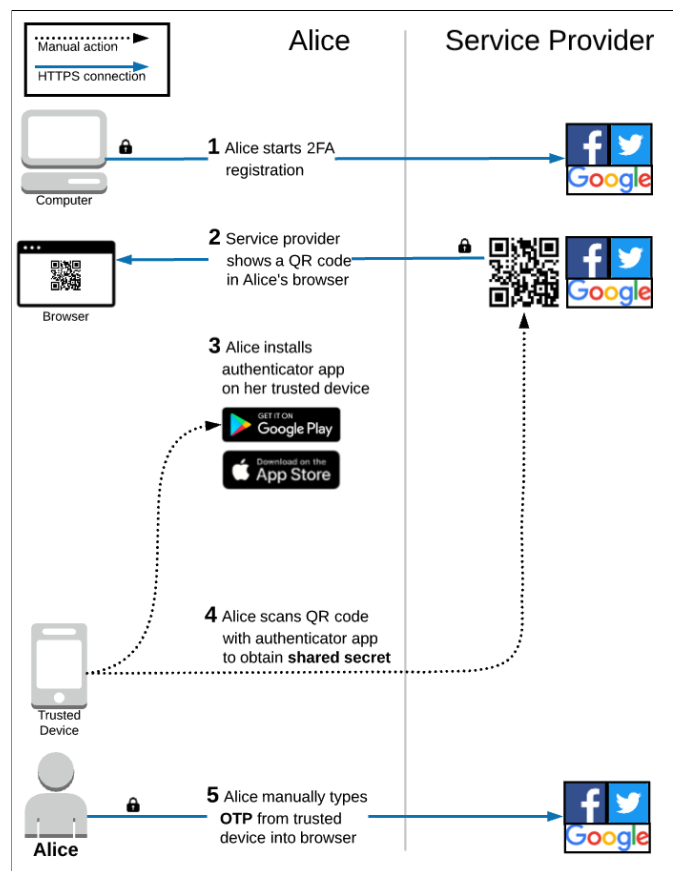


Figure 2.1: Standard workflow for a user to enable TOTP on their account for the first time.

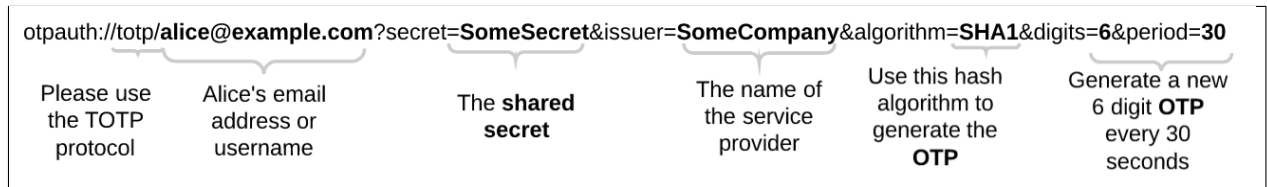


Figure 2.2: Contents of a TOTP setup QR-code [60].

After Alice starts the TOTP registration process (Figure 2.1, step 1), the service provider displays a setup QR-code (Figure 2.1, step 2). If this is the first time that Alice has used TOTP 2FA on any account, they will need to install a TOTP app on their trusted device (Figure 2.1, step 3), such as a phone or tablet. For brevity, I will refer to a user's phone throughout the rest of this dissertation. Alice only needs to install a TOTP app a single time. Often, the service provider will recommend one or two specific TOTP apps to the user. However, any app that implements the TOTP algorithm will generate the same OTP codes, so the choice of which TOTP app to use is mainly based on user preference.

Once the TOTP app is installed, Alice uses it to scan the setup QR-code (Figure 2.1, step 4) generated by the service provider. The contents of the setup QR-code follows the defacto standard defined by Google [60]. Figure 2.2 shows an annotated example of the URI encoded in the setup QR-code.

The three most important fields included in the setup QR-code are:

1. **Website:** the name of the website/service on which the user is currently enabling TOTP;
2. **Username:** the user's account username; and
3. **Secret:** a static, random secret generated by the service that is unique to the user's account.

The TOTP app uses the *website* and *username* to visually indicate in the app which OTP go with which account. The *secret* is used by the app to generate a 6-digit OTP locally on Alice's device. The OTP is manually entered into the browser (Figure 2.1, step 5) to complete the registration process. During future authentication sessions, Alice simply opens the TOTP app, locates the correct 6-digit code, and enters it into the browser when prompted.

Because the TOTP secret is static, anyone with possession of the secret can generate valid OTPs for the user's account. This places a burden on the website developer to save the TOTP secret securely, such as encrypting it with a hardware security module (HSM) to protect against exposure in the event of a data breach. Attackers could also steal the TOTP secret directly from the user's device, but that typically requires physical possession of the device. In practice, it is probably more likely that the user would simply lose access to the TOTP secret; people lose their devices, buy new ones, or uninstall their TOTP apps.

Without the shared secret, the user cannot generate the OTPs required to authenticate and could face account lockout.

# Chapter 3

## Related Work

Time-based One-Time Passwords (TOTP) is one of many 2FA mechanisms frequently deployed in the real world. Each method of 2FA has tradeoffs in terms of security, privacy and usability. Therefore, online services rely on multiple authentication mechanisms in practice [1].

This chapter outlines prior work that has investigated various methods of 2FA, including the most popular methods of 2FA deployed in the real world and various solutions proposed in research projects. Additionally, it highlights related research on various account recovery mechanisms (a.k.a “fallback authentication” and “recovery mechanisms”) for accounts protected with 2FA and the lack of prior work focused on the backup strategies of TOTP users.

### 3.1 2FA Adoption

2FA has consistently proven to drastically increase the security of online accounts [91, 106, 25], but industry adoption rates have been alarmingly low in recent years. Companies are often guarded about the 2FA adoption rate among their active user bases, which has forced researchers to devise ways to estimate usage [81]. Over the years, some companies have been slightly more forthcoming with this data, which I have documented in a public spreadsheet that is partially replicated in Table 3.1. Notable examples include Twitter (2.3% as of July 2021), GitHub (15.5% as of August 2021), NPMjs (9.6% as of March 2020), Google (10% as of 2018), and Dropbox (<1% as of 2016).

Date Reported	Adoption Rate (%)	Company/Software	Quote
2023-02-17	21.62	ReportURI	“We currently stand at 21.62% of users with 2FA activated, so a nice rise in the relative number and quite a significant bump in the absolute numbers!”
2022-12-07	95	Apple ID	“Apple introduced two-factor authentication for Apple ID in 2015. Today, with more than 95 percent of active iCloud accounts using this protection...”
2022-02	22	Azure Active Directory	“22% of Azure Active Directory uses strong authentication (MFA and passwordless).”
2021-08-24	15.5	GitHub	“You can update GitHub’s number to 15.5.”
2021-07-14	2.3	Twitter	“2.3% of active Twitter accounts had at least one 2FA method enabled on average during July thru December 2020.”
2020-03-10	10	Eve Online	“As of now, for the last 30 days, over 10% of logins have 2FA enabled :)”
2020-03-10	9.6	NPM	“I just checked today and maintainers with 2FA enabled is now at 9.60%.”
2018-12-14	52	OneLogin	“52% of OneLogin’s customers use 2FA.”
2018-09-24	1.7	Azure Admins	“The rate increased from 0.7% in 2017 to 1.7% in 2018. Yes, it doubled, but it is still terrible.”
2018-08-22	20	Office 365 admins and users	“Only 20 percent of organizations use MFA for admins and users.”
2018-01-17	~10	Google	“Less than 10 percent of active Google users use second factor authentication.”
2016-02-11	<1	Dropbox	“Less than 1% of Dropbox users had 2FA enabled at the time.”

Table 3.1: Rates of 2FA adoption recorded by Conor Gilsean. Sources available at: [allthingsauth.com/2fastats](https://allthingsauth.com/2fastats).

Many researchers have investigated various strategies for increasing 2FA adoption. Redmiles et al. [89] conducted an interview and participatory design study ( $n = 12$ ), in which they documented participant feedback on existing 2FA messages and proposed new messaging for prompts to encourage 2FA adoption. Both Al Qahtani et al. [3] and Albayram et al. [4] studied the use of video to increase 2FA adoption and found that, depending on the design, videos increased adoption up to 27%. Conducting experiments in collaboration with Facebook, Golla et al. [48] found that changes to the design and messaging of security prompts that considered users' motivation, mental models, and concerns increased adoption rates of 2FA among Facebook users by up to 30%. Das et al. [20] found that timely communication of risks and benefits was a critical to successfully deploying 2FA at a large company.

Previous work studied various aspects of 2FA deployments at universities. In general, university populations found 2FA to be annoying yet usable [110, 17, 29], and were more willing to endure usability challenges with 2FA for systems they deemed as sensitive [2]. Dutson et al. [29] recommended a host of usability improvements to DuoMobile, the most common 2FA system deployed at their universities. Reynolds et al. [95] also recommended to give careful consideration to session timeout durations and to highlight the use of the "Remember Me" option to reduce the frequency of 2FA challenges, a key usability issue. Studies with non-university students have corroborated the finding that users generally report 2FA as usable [23, 92].

Though it was a small effect size, Colnago et al. [17] found that users were more likely to positively respond to 2FA system if they voluntarily opted in. However, most users do not understand the security benefits of 2FA [58, 90, 22], which contributes to low adoption rates. Therefore, many researchers recommended that organizations strongly consider mandating 2FA usage. Recently, there appears to be a paradigm shift throughout industry and several large sites (e.g., Google [70] and Salesforce [7]) are requiring users to adopt 2FA.

It is well established that a major road block to 2FA adoption in general is the fear of losing physical authenticators [94, 19, 16, 36, 67, 77, 21]. FIDO2 [37], which is supported by all major browsers, provides an authentication mechanism that relies on physical possession of a device and is resistant to phishing attacks, but research has consistently found that users are concerned about losing their authenticator and getting locked out of their accounts [94, 19, 16, 36, 67, 77]. After evaluating 12,500 crowd-sourced comments about 5 prevalent 2FA mobile apps, Das et al. [21] called for improvements to backup and recovery mechanisms to eliminate account lockout concerns. Because of the real world risk of users locking themselves out of their accounts, and the associated real world costs of helping those users regain access to their accounts, websites commonly implement "fallback authentication" mechanisms that do not require physical possession of a device, such as security questions, email, and SMS.

## 3.2 2FA and Account Recovery

This section outlines the various account recovery mechanisms (a.k.a “fallback authentication” and “recovery mechanisms”) associated with common implementations of 2FA.

### 3.2.1 Secret Questions

One of the most common “fallback authentication” mechanisms, often used to reset a forgotten password, is the use of “knowledge-based recovery questions” [84]. Schechter et al. [98] conducted a lab study in 2009 to determine whether participants could remember their answers to security questions and whether their answers were easily guessable. The authors concluded that security questions are “significantly weaker than passwords” and are likely not reliable enough for use as authenticators. These findings were bolstered in 2015 by Bonneau et al. [12], who analyzed real-world data from Google that included 100s of millions of answers to secret questions and millions of account recovery attempts. The authors concluded that “it appears next to impossible to find secret questions that are both secure and memorable” and strongly recommended against using security questions as a standalone recovery mechanism since significantly more reliable methods are available (SMS and email, according to their data). Even more recently in 2019, Doerfler et al. [25] analyzed more Google data and found that users could only provide correct answers to security questions 78% of the time.

### 3.2.2 Email

Due to the security and usability issues with “knowledge-based recovery questions” discussed in Section 3.2.1, websites often rely on email to complete account recovery. Li et al. [66] found that allowing password recovery via email alone was nearly ubiquitous among the Alexa Top 500 and stressed that delegating recovery to email providers makes user inboxes a single point of failure. Doerfler et al. [25] analyzed large, real world data sets collected by Google’s authentication systems and stated that sending an OTP to a secondary email address encountered significant usability and security issues; users were only able to pass the challenge 22% of the time and attacks flagged as phishing attempts were able to prove they had access to the secondary email accounts 8% of the time.

Raponi and Di Pietro [85] proposed Maildust, a password recovery scheme that protects against internal attackers at email providers by requiring multiple recovery email addresses. Ultimately, Maildust suffers from the same security and usability issues that [25] observed: users are likely to forget their secondary email addresses and attackers have demonstrated their ability to access secondary emails.

### 3.2.3 SMS

Doerfler et al. found that SMS 2FA enabled on Google accounts was highly effective against automated attacks by bots, but only prevented 76% of targeted attacks [25]. SMS 2FA is appealing to many people because of its familiarity and its “ease of recovery” if the phone is lost: simply get a replacement SIM card from your service provider. However, this highlights that SMS OTP does not satisfy a strict definition of “something you physically have,” which leads to a host of security vulnerabilities. SMS 2FA is not considered a secure authentication mechanism due to the ease of hijacking phone numbers [18, 65]. After observing the prevalence of recycled phone numbers actively associated with the accounts of the previous owner, Lee and Narayanan [64] concluded that websites/services “...should no longer equate a correctly-entered SMS passcode with successful user authentication.”

### 3.2.4 Time-based One-time Passwords (TOTP)

RFC6238 [68] defines the technical algorithm for the client and server to use a shared secret to generate and validate a deterministic one-time password (OTP)—usually a numeric code—during authentication. However, there is no standard for how to back up the shared secret on the client. Without the TOTP shared secret, the user cannot generate the OTPs required to authenticate and could face account lockout.

Das et al. [21] evaluated 12,500 crowd-sourced comments about five prevalent 2FA mobile apps<sup>1</sup> and found many users complained about account lockout. One key recommendation the authors made was to improve backup and recovery mechanisms to address concerns about disaster recovery. It is common for people to leave online reviews only in the extreme cases (something terrible or something amazing) [54]. Therefore, it could have been that a vocal minority of TOTP users were complaining about lockout while most users were fine, or that account lockout was a rampant problem among TOTP users. There was no publicly available data measuring how often TOTP users actually faced account lockout and which backup mechanisms TOTP users actually leveraged in practice. The existing backup and recovery options for TOTP users were not well explored in the literature, which is a necessary starting point from which to design improvements. This dissertation fills that gap and explores the functionality of existing backup mechanisms for TOTP users and how they get used in the real world (see Chapter 5).

In my first project (see Chapter 4), I analyzed the top 22 TOTP Android apps and found numerous security and privacy issues. My work in this area has already inspired other researchers to continue investigating similar security and privacy aspects of TOTP apps [55].

Researchers have also focused on other aspects of TOTP, including how the TOTP secret is stored on the physical device [82, 78], other insecure development practices [55], replay attacks [10], and the overall usability of the TOTP scheme [92]. I have also written previously about phishing and social engineering attacks on TOTP [45].

---

<sup>1</sup>The mobile apps included Authy, Microsoft Authenticator, Duo Mobile, Google Authenticator, and Okta.



### 3.2.5 Push 2FA

Push notification 2FA (“Push 2FA”) is a generic authentication technique that relies on public key cryptography. Typically, the user will install an app specific to the website or service on which they have an account (e.g., Microsoft Authenticator). A unique key pair is generated for the user’s device; the public key is sent to the website/service’s server while the private key is stored securely on the user’s device. To authenticate, the app on the user’s phone receives a notification alerting the user to the authentication attempt and asking for approval. Under the hood, the app has received a challenge from the website/services’s server. The user can approve the authentication attempt with a single click, which will actually use the private key to sign the challenge and return it to the website/service.

Duo, a commercial service that provides Push 2FA, documents that their Duo Mobile product follows this architecture [27].

While this authentication mechanism has some usability improvements to the manual OTP entry required in most TOTP implementations, Push 2FA is still vulnerable to social engineering attacks. Additionally, there is typically no way to export the private key, which leaves no realistic backup options. Without an alternative authentication mechanism also enabled, the user could face account lockout.

### 3.2.6 FIDO2

The FIDO2<sup>2</sup> collection of specifications is supported by all major browsers<sup>3</sup> and uses public key cryptography to provide a secure authentication mechanism that is resistant to phishing attacks. MacGregor [69] highlighted that Android-based FIDO2 authenticators do not provide easy recovery from loss. Many researchers have conducted usability studies specifically on hardware-based (i.e., security keys) and phone-based FIDO2 authenticators [94, 19, 16, 36, 67, 77] and consistently found users report that they were concerned about losing their authenticator and getting locked out of their accounts.

The recommendation of the FIDO2 project is to preregister a backup authenticator. This is highly unsatisfying because backup keys should be stored in a safe place and not used on a regular basis, but preregistering the backup key on each site requires regular access to the backup key. Yubico recently proposed an extension to FIDO2<sup>4</sup> that allows a primary and backup security key to be cryptographically associated such that the backup key can be left in a secure place while the primary key is used to register new accounts and authenticate to existing accounts. In the event the primary key is lost, the backup key is immediately usable for authentication at websites where the primary key was registered, and a new backup key will be provisioned once the user obtains a replacement. Yubico collaborated with Frymann

---

<sup>2</sup><https://fidoalliance.org/fido2/>

<sup>3</sup>All of Chrome, Edge, Safari, and Firefox support the WebAuthn specification, which is part of the FIDO2 project.

<sup>4</sup><https://web.archive.org/web/20210605113624/https://www.yubico.com/blog/yubico-propose-s-webauthn-protocol-extension-to-simplify-backup-security-keys/>

et al. [42] to evaluate Asynchronous Remote Key Generation (ARKG), the new cryptographic primitive required to make the proposed scheme work.

In recent years, there has been a massive push by industry to increase the adoption of FIDO2 strong authentication. Significant work has been done to turn the existing devices that people have, such as phones and tablets, into FIDO2 authenticators. Relaxing the security requirements slightly and implementing FIDO2 in software creates the opportunity to backup the private keys required for authentication. Microsoft [59], Apple [8], and Google [115] call this strategy *passkeys* and are heavily investing in its adoption. The private keys are encrypted locally on the user’s device and are stored remotely in the users’ cloud storage account at the respective service provider. This architecture alleviates the need for each website online to implement their own recovery mechanisms, but it does create a single point of failure by centralizing backups. For example, suppose a user has all of their passkeys backed up into their Google account and loses their phone. Assuming that the user does not have a secondary device, then it is likely that the FIDO2 key that was required to log into their Google account is also lost. Therefore, the user must rely on weaker forms of authentication, such as passwords, email, and SMS. Attackers are likely to focus their efforts on these centralized backup repositories.

### 3.2.7 Account Recovery Codes

Account recovery codes are random, one-time use passwords (OTPs) that are typically provided to the user when they are enabling 2FA on their account. In general, prior research into the efficacy, security, and usability of account recovery codes is understudied. However, Doerfler et al. [25] analyzed a large, real-world data set from Google and observed that only 7% of users who had 2FA enabled were able to pass the recovery code challenge. The cause of this failure is unclear, but the authors speculated that users had difficulty locating their recovery codes.

### 3.2.8 Social Authentication

The SoK paper by Alomar et al. [5] provides a thorough overview of many social authentication schemes. Most social authentication schemes rely on the user knowing something *about* the social network. A prominent example is Facebook’s recovery feature that required users to identify pictures of their friends. However, most of these approaches are simply a specific type of secret question and do not provide a meaningful level of security (see Section 3.2.1).

A more compelling area of social authentication involves proposals that require active participation of individuals *within* the social graph. Brainard et al. [14] argued that “someone you know” could be considered a fourth factor of authentication and outlined a system that maintains a pre-registered list of Helpers who can assist an Asker recover their account. Schechter et al. [99] explored a system that allowed users to recover their Microsoft account by contacting 3 out of 4 people previously identified as trusted contacts (trustees) via email. Facebook implemented a similar strategy in their Trusted Contacts feature, where a user

can receive codes from three pre-registered friends to regain access to their account [34, 35]. Shropshire and Menard [102] proposed a social recovery system in which users send a video recording of themselves to preregistered trustees to unlock their mobile phone if they forgot their lock screen PIN; this could be easily adapted to authentication scenarios for online accounts.

All of these approaches rely on a trusted central server that also acts as the authentication server. This means that the central server could act maliciously and enroll someone as a helper/trustee without their approval or knowledge. In fact, [14] went as far as to state that they “assume the trustworthiness of the enrollment process, and the system administration and server”. Social engineering is a primary concern for social authentication mechanisms. Facebook Trusted Contacts has proven vulnerable to phishing attacks [5]. Schechter et al. [99] evaluated their scheme against social engineering specifically.

### 3.2.9 Secret Sharing

Secret sharing schemes generate unique secret shares from an original secret such that none of the secret shares divulge information about the original secret. Threshold secret sharing schemes, like Shamir’s secret sharing [100] allow the reconstruction of the original secret using only a predefined minimum threshold of secret shares. For example, in a 3-of-5 scheme, 5 total shares would be generated, but the original secret can be reconstructed using any 3 shares.

Though not common among the general population, secret sharing is widely used today in practice. I define two flavors of secret sharing schemes: those that store shares on  $n$  remote servers, and those that store shares with  $n$  people.

**N Servers** There are many existing and proposed systems that use secret sharing for authentication or to backup data by storing shares on remote servers.

Raponi and Di Pietro [85] proposed Maildust, a password recovery scheme that protects against internal attackers at email providers. Instead of sending a password reset link to a single email account, Maildust sends secret shares to multiple preregistered email accounts. Only by combining these secret shares can the user reset their password. While this does somewhat address the risk of internal attackers at email providers, it is likely that people will have multiple email accounts with the same provider. This creates a poor user experience that requires the user to authenticate to each of the  $n$  servers independently.

Work by Chen et al. [24] cleverly uses Secure Multipart Computation (SMPC) to allow  $n$  servers to collaboratively authenticate a user with only a single authentication action, rather than  $n$  separate authentication actions, one to each server. To enable this, the authors describe and implement TLS-in-SMPC, which allows the  $n$  servers to appear as a single TLS client to an unmodified TLS server, such as an email gateway for sending or telephony API to send one-time passwords (OTPs). While this clever work significantly improves the user experience, the single authentication that is still required still relies on the same common authentication mechanisms. The authors propose backing up secrets to a system that uses

“N-for-1 authentication,” but consider the case where a user loses their phone and is trying to recover that secret on the remote server. The user cannot leverage strong authentication mechanisms that require physical possession (e.g., FIDO2) if their secrets were stored on the lost phone. Instead, the user must rely on weaker authentication methods that do not require physical possession (e.g., passwords, email, SMS).

In their work, Chen et al. [24] highlight many other commercial products that leverage secret sharing by storing shares on remote servers. All of these systems face the same account recovery conundrum since the user has to authenticate to a remote system.

**N People** An alternative approach to storing secret shares on remote servers is to distribute the shares to people who you know in the real world; these people act as Recovery Contacts and can help you recover if you lose all of your personal devices on which your secrets are stored. There are several real world examples of this approach to secret sharing. Hashicorp Vault leverages Shamir Secret Sharing to seal and unseal secret vaults [50]. SatoshiLabs is working on a specification to standardize SS backup implementations [103], which has been implemented in Trezor [75] and Keystone [61] cryptowallets. These cryptowallets warn people to write their secret shares on paper and to avoid digitizing them by taking photos or storing them on a computer because of the security risk of breaches and attacks.

## Chapter 4

# Security and Privacy Analysis of Backup Mechanisms in TOTP Apps

As reviewed in Chapter 2, a user could face account lockout if they lose access to their TOTP app. To combat this usability nightmare, many TOTP apps provide custom backup mechanisms to help users recover from device loss. These backup features are critical usability enhancements, but it was understudied how they impacted the security and privacy of the millions of people who use these TOTP apps on a daily basis.

Thus, this chapter presents published work<sup>1</sup> on which I was the primary author that aimed to comprehensively investigate the security and privacy issues that exist in the backup mechanisms of the most used TOTP apps.

Our research questions included:

- RQ1** What personal information, if any, is leaked to the company that develops the TOTP app, or other third parties, as a result of using the TOTP backup mechanisms?
- RQ2** What is the risk of an attacker obtaining a TOTP backup?
- RQ3** What is the risk of an attacker compromising the TOTP secret(s) stored within an obtained TOTP backup?

To answer these questions, we identified all general purpose Android TOTP apps in the Google Play Store with at least 100k installs that implemented a backup mechanism ( $n = 22$ ). For each app, we manually registered TOTP accounts and exercised the available backup mechanism(s) while recording plaintext network traffic. If we determined that encryption beyond TLS was used, we performed a cryptanalysis of the app to determine how the implementation and/or use of cryptography impacted the security of the TOTP backup.

While TOTP 2FA is often billed as a security enhancement relative to passwords and SMS-based 2FA [26], our analysis found that most backup strategies end up placing trust in

---

<sup>1</sup>Published at USENIX 2023 [47], a top-tier, peer-reviewed conference.

the same technologies that TOTP 2FA is meant to supersede: passwords, SMS, and email. The most commonly-supported backup mechanisms were syncing encrypted TOTP backups automatically to the cloud and leveraging the built-in Android Auto Backup system. Alarmingly, two apps sent the plaintext TOTP secrets to the app developer. Apps that encrypted TOTP backups had a wide range of serious vulnerabilities. Several apps ( $n = 4$ ) sent both the encrypted backup and the encryption key (or the password from which it was derived) to the app developer, allowing them to decrypt the backup and read its content. Nearly all apps that encrypted TOTP backups derived a key from a user-provided password, but most also had severely inadequate password policies and/or used weak methods of deriving a key from the password. As a result, the ciphertexts were vulnerable to trivial offline attacks. More than half ( $n = 12$ ) of the apps we analyzed allowed the user to manually or automatically create plaintext TOTP backups, but few of those apps warned the user about the risk of doing so. Our contributions are as follows:

- We present a methodology for evaluating the security and privacy properties of backup mechanisms in TOTP apps, using both dynamic analysis and cryptanalysis.
- We show that many popular 2FA apps used vulnerable security mechanisms, thereby exposing TOTP secrets.
- We show that many popular TOTP apps leaked user information, including the names of the websites/services they use and their account usernames on those platforms.

## 4.1 Methods

In this section, we explain how we selected the 22 TOTP 2FA Android apps to analyze and the systematic procedure that we used to analyze each app.

Throughout the rest of the paper, we use the term *plaintext* to refer to data that is not end-to-end (e2e) encrypted before leaving the app. We use the term *encrypted* to indicate that the app applied e2e encryption prior to transmission (i.e., regardless of whether TLS was used for transmission; all apps tested used TLS when transmitting backup data).

While there are myriad ways an attacker can compromise a device that is in their physical possession, we consider these attacks out of scope. Instead, our analysis focused on the threat model of an attacker obtaining the data contained in a TOTP backup (e.g., the secret, label, and issuer) once it leaves the local device. We assume that the Android device is free of malware and has the latest security updates applied.

### 4.1.1 App Selection

In November and December 2021, we identified as many TOTP apps in the Google Play Store as possible. To start, we created a list of core search terms that we knew from personal experience would return TOTP apps (e.g., “totp”, “2fa”, “two factor”, “mfa”,

and “multifactor”). We entered each core search term into the search box on the Google Play Store and incorporated the top 5 auto-completion suggestions.<sup>2</sup>

We queried the Google Play Store for each unique search term and downloaded the metadata for the top 30 apps displayed in each query result set. Removing duplicates yielded 546 apps. We further narrowed the candidate pool by excluding 193 apps that had fewer than 100,000 installs and 263 apps whose description did not contain any of the final search terms. The remaining 90 apps were each reviewed manually to determine whether they were, in-fact, TOTP 2FA apps.

We focused exclusively on apps whose primary focus was 2FA and could be used on any site that supports TOTP. We excluded any apps that only worked on a specific service (e.g., Blizzard Authenticator). For purposes of practicality and scope, we excluded multipurpose apps that offer TOTP support amongst other functionality (e.g., password managers). Analyzing the 22 TOTP apps that satisfied our criteria took a significant amount of time and effort. While many TOTP apps include some cryptographic features that are outside of our scope (e.g., push 2FA, encrypting local storage), many password managers use cryptography much more extensively than TOTP apps, which would make analysis even more complex.

### 4.1.2 App Analysis

For each app in our data set, we downloaded the Android Package (APK) file from the Google Play Store using a non-rooted Pixel 3a phone, the Android Debug Bridge (ADB), and a custom shell script.<sup>3</sup> We analyzed each APK using the three phases described in the following subsections.

#### 4.1.2.1 Exploring the App

The first step of analysis was to explore the app and document its various features and settings. We recorded whether the app required any personal information to use the app at all. For example, the *Twilio Authy* app required the user to provide an email address and prove control of a phone number.

We also answered a range of other questions that could be determined by using the app. What personal information, if any, is required to enable/utilize the backup mechanism? What backup mechanisms does the app support (e.g., remote backups, manual exports, transfer via QR code, etc.)? If remote backups are available, where are they stored (e.g., the developer’s cloud service, Google Drive, etc.)?

If the backup mechanism required creating a password, then we attempted to manually determine the password policy, including minimum required length and use of a block list.

Finally, we created a customized checklist that enumerated exactly which actions to take within the app and which data to enter (e.g., phone numbers, email addresses, and passwords) while recording the network traffic. Each checklist included a core set of steps designed to

---

<sup>2</sup>See Section 4.4 for supplemental material available online.

<sup>3</sup><https://github.com/blues-lab/getapk>

replicate the real world actions that a user would take when first using the app, enabling its backup mechanism, and eventually executing the recovery process. We believe that a reasonable user executing these steps would not expect any of their personal information to be sent remotely in plaintext, unless they were informed.

We specifically documented details about each app’s recovery workflow so that we could determine which attack vectors were available to an attacker attempting to impersonate a user and obtain the TOTP backup remotely.

#### 4.1.2.2 Capturing & Reviewing Network Traffic

After becoming familiar with the app, we followed the customized checklist to exercise the app’s backup and recovery mechanisms in a controlled environment and noted which information, if any, was sent remotely. The goals of this phase were to (1) identify any plaintext TOTP fields in backups that were transmitted remotely and (2) record any fields in the TOTP backups that appeared to be encrypted.

We installed each app on a Pixel 3a phone running a custom version of Android 9 (initially developed in prior research [113, 114, 93, 86] and commercially maintained by AppCensus<sup>4</sup>) that recorded plaintext network traffic to file.<sup>5</sup> A variety of open source tools for collecting network traffic (e.g., mitmproxy,<sup>6</sup> Magisk,<sup>7</sup> and Frida<sup>8</sup>) can be used to verify our results and, we believe, reproduce our findings from scratch.

We ensured that each phone had a SIM/phone number and a registered Google account. For reference and completeness, the phone’s screen was recorded as an mp4 video using scrcpy<sup>9</sup> during the network logging session. The ability to retroactively review the specific actions that led to a specific transmission proved to be an invaluable time-saving resource.

As mentioned previously, the most common method of setting up TOTP 2FA is for the user to scan a QR code using the TOTP app. We followed the de facto standard [60] to create two QR codes that encoded specific values for the issuer, label, and secret fields. Using these QR codes throughout our analysis allowed us to check whether these known values appeared in the network traffic generated by each app.

After recording, we reviewed the network traffic to identify which request/response calls were responsible for sending the TOTP backup to the remote storage service, if any. We checked whether the network traffic contained any known values, such as passwords entered to enable the backup mechanism and any of the fields encoded in our custom QR codes. Finally, we documented the value of any field that appeared to be encrypted so that it could be further analyzed in detail.

---

<sup>4</sup><https://appcensus.io>

<sup>5</sup>See Section 4.4 for supplemental material available online.

<sup>6</sup><https://mitmproxy.org/>

<sup>7</sup><https://github.com/topjohnwu/Magisk>

<sup>8</sup><https://frida.re/docs/android/>

<sup>9</sup><https://github.com/Genymobile/scrcpy>



### 4.1.2.3 Performing Cryptanalysis

If the collected network traffic contained any encrypted fields, we performed a cryptanalysis of the app to determine which cryptographic primitives were used to create the ciphertext in the TOTP backup and how they were configured. Understanding these details is critical to assessing the overall security of the app’s backup mechanism because any attacker with access to the ciphertext would attempt to learn the same information to launch an offline attack.

For apps that were not open sourced, we decompiled the APK using the jadx decompiler.<sup>10</sup> Analyzing obfuscated code was often complex. We identified potentially-relevant code by searching for static strings used in crypto libraries (e.g., AES, PBKDF2), compared function signatures to open source crypto libraries (e.g., Bouncy Castle<sup>11</sup> and libsodium<sup>12</sup>), and manually refactored the code to improve readability.

Our cryptanalysis consisted of two main phases. In the first phase, we reviewed the code to determine how the app obtained the encryption key used to generate the ciphertext in the TOTP backup. The most prevalent backup architecture among the apps we analyzed involved deriving a symmetric encryption key from a user-provided backup password. Therefore, we documented the key derivation function (KDF) that was used and how it was configured, including whether or not a salt was used and, if so, whether it was random or static.

In the second phase, we focused on determining the cryptographic primitives used to produce the ciphertext, including which encryption ciphers were used, their modes of operation, and the method of authentication, if any. Most apps did not provide any integrity over the ciphertext, so we recorded any custom heuristics used to determine whether or not the correct backup password was provided during recovery. For example, some apps checked whether the decrypted backup was a valid JSON object. These are the same heuristics that an attacker would use in an offline attack.

Many of the apps that we analyzed implemented multiple features that relied on various methods of cryptography. Therefore, we could not assume that our observations and assumptions about the cryptography used to generate the ciphertext in the TOTP backup were correct. To prove that we had identified the correct cryptographic primitives and configurations, we implemented the app’s decryption process in a separate script<sup>13</sup> that accepted the following inputs: (1) the ciphertext and IV from the network traffic; if applicable, (2) the password that we chose when enabling the backup mechanism, and (3) any salt passed to a KDF. We manually verified whether the decryption process was correctly implemented by looking for known values in the resulting plaintext, such as the TOTP secret from our custom QR codes.

---

<sup>10</sup><https://github.com/skylot/jadx>

<sup>11</sup><https://www.bouncycastle.org/>

<sup>12</sup><https://libsodium.org>

<sup>13</sup>See Section 4.4 for supplemental material available online.

Abbreviated Name	APK id@version	Installs	Backup Mechanisms							
			QR Codes	Cloud Sync		File Export		Sharing		Android Backup
				Plaintext	Encrypted	Plaintext	Encrypted	Plaintext	Encrypted	
Google Authenticator	com.google.android.apps.authenticator2@v5.10	100M+	Y	-	-	-	-	-	-	-
Microsoft Authenticator	com.azure.authenticator@v6.2204.2757	50M+	-	-	Y*	-	-	-	-	-
Duo Mobile	com.duosecurity.duomobile@v4.15.0	10M+	-	-	Y	-	-	-	-	-
Twilio Authy	com.authy.authy@v24.8.5	10M+	-	-	Y	-	-	-	-	-
Latch	com.elevenpaths.android.latch@v2.2.4	5M+	-	Y	-	-	-	-	-	-
LastPass Authenticator	com.lastpass.authenticator@v2.5.0	1M+	-	-	(Y)	-	-	-	-	-
2FAS	com.twofasapp@v3.11.0	1M+	-	Y	Y*	Y	Y	Y	Y	-
Yandex.Key	ru.yandex.key@v2.7.0	1M+	-	-	Y*	-	-	-	-	-
FreeOTP Authenticator	org.fedorahosted.freeotp@v1.5	1M+	-	-	-	-	-	-	-	Y
Authenticator	com.pixplicity.auth@v1.0.6	500k+	Y	-	-	-	-	Y	Y*	-
Salesforce Authenticator	com.salesforce.authenticator@v3.8.5	500k+	-	-	Y*	-	-	-	-	-
Code Generator	net.codemonkey.otpgeneratorapp@v6.1	500k+	-	-	-	Y	-	-	-	Y
TOTP Authenticator	com.authenticator.authservice2@v1.89	100k+	Y	-	Y*	-	Y*	Y	Y	Y
Aegis Authenticator	com.beemdevelopment.aegis@v2.0.3	100k+	-	-	-	Y	Y	Y	Y	Y
Auth0 Guardian	com.auth0.guardian@v1.5.3	100k+	-	-	-	-	-	-	-	Y
App Authenticator	authentic.your.app.authenticator@v1.5	100k+	Y	-	-	-	Y*	Y	-	Y
andOTP	org.shadowice.flocke.andotp@v0.9.0.1-play	100k+	Y	-	-	Y	Y^	-	-	Y
Zoho OneAuth	com.zoho.accounts.oneauth@v2.1.0.5	100k+	-	-	Y*	-	-	-	-	-
Authenticator Pro	me.jmh.authenticatorpro@v1.15.10	100k+	-	-	-	Y	Y	-	-	-
SAASPASS	com.solidpass.saaspas@v2.2.28	100k+	-	Y	-	-	-	-	-	-
Authentic Password	authentic.password.authenticator.pro@v1.3	100k+	Y	-	-	-	-	Y	-	Y
Mobile Authenticator	authenticator.mobile.authenticator@v1.7	100k+	Y	-	-	-	-	Y	-	Y
	TOTAL apps	-	7	3	9	5	6	7	4	9
	TOTAL installs	181.5M+	101M+	6.1M+	73.7M+	1.8M+	1.5M+	2M+	1.7M+	2.2M+

Table 4.1: Overview of the backup mechanisms supported in each app. **Y\*** indicates that there is a serious security flaw in the implementation and/or usage of cryptography (see Section 4.2.3). **Y^** indicates support for multiple types of encrypted file exports (see Section 4.2.3.4). Values in parentheses were obtained from documentation and observation only (see Section 4.3.4).

## 4.2 Results

In this section, we explain the various backup mechanisms available in the apps that we analyzed and discuss the security and privacy implications of each.

### 4.2.1 Backup without the Network

Like the *de facto* standard [60] for initially transferring TOTP data from the website/service to the TOTP app, 7 apps (see Table 4.1) leveraged QR codes to allow users to optically transfer TOTP secrets between devices without the risk of sending data over the network (RQ2). Of these, several bury this feature below multiple menu layers, decreasing the odds that users will find the feature unless specifically looking for it.

Manual export via QR code is the only backup mechanism supported by Google Authenticator, the app with the most installs ( $n = 100M+$ ) by a factor of 2 or more. Google stated that balancing security and usability was a specific motivation when it rolled out support for this feature and highlighted that “...physical access to [the] phone and the ability to unlock it” is required [51]. Google Authenticator also records an audit log that allows users to detect suspicious secret transfers.

While backups via QR code prioritize confidentiality and integrity, the mechanism falls short on availability. Scanning a QR code with another personal device is inherently a manual activity. Users must own one or more secondary devices, know their location, and physically transfer TOTP data from their primary device to their secondary device(s) each time they add a new account. We believe it is likely that many users will not perform this backup ritual reliably because prior work has found that people are often unsuccessful at regularly creating manual backups for their devices in general [87]. While periodic nudges or other reminder techniques may increase the manual backup rate, many users could face account lockout when they inevitably forget to backup some TOTP secrets and actually need to recover.

The threat model for some users may demand the security benefits of using QR codes to transfer TOTP secrets between devices, but we suspect that most users will prefer more automated solutions. To enable any type of automated backup system, some data must inevitably be sent over the network.

### 4.2.2 Remote Backups *without* Encryption

As shown in Table 4.1, more than half (12) of the apps in our dataset, comprising almost 8 million installs, are capable of backing up the TOTP data in plaintext. The majority of these apps (10) supported sharing plaintext backups directly via the standard Android sharing menu (e.g., via email, SMS, etc.), copying the plaintext backup to the clipboard, and/or exporting the plaintext backup to the device file system where it could be shared like any other file.

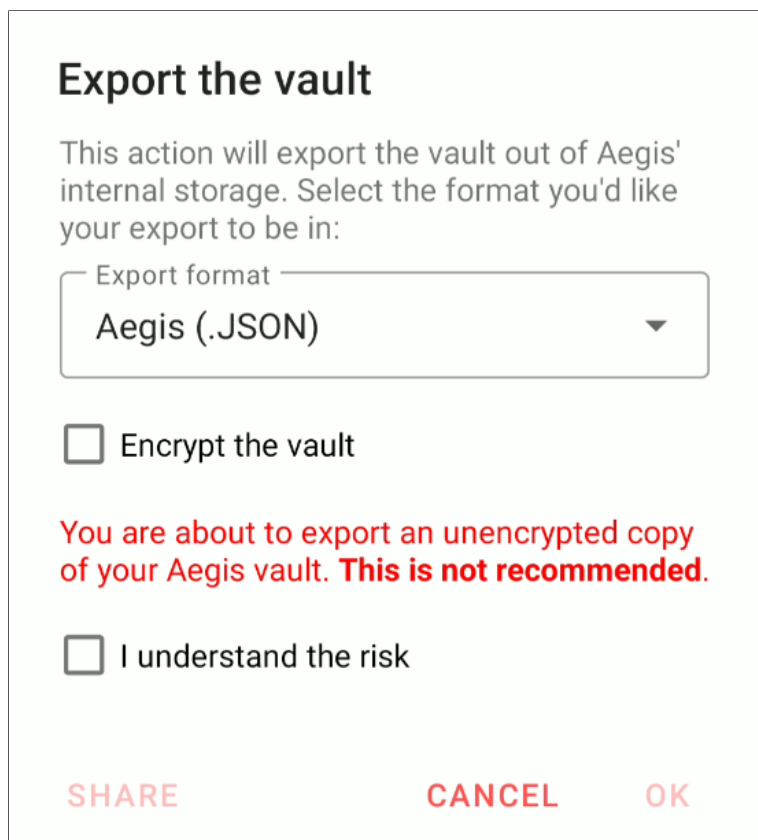


Figure 4.1: Plaintext warning in the *Aegis Authenticator* app.

Just 3 apps supported plaintext cloud sync, which sent all of the TOTP fields (secret, label, issuer) to the cloud in plaintext. The *Latch* app automatically sent all TOTP fields to Latch servers in plaintext with no option for users to opt out. This was the only backup mechanism the *Latch* app provided to its 5M+ users. Plaintext cloud syncing was also the only backup mechanism supported by the *SAASPASS* app; if enabled, it backed up the TOTP fields in plaintext via the XMPP protocol. Though the feature was optional and off by default, *SAASPASS* regularly prompted the user to enable it. Similarly, the *2FAS* app prompted users to enable remote backups to Google Drive, which were in plaintext by default. The app did support an additional option to encrypt remote backups, but a UX flaw still leaked plaintext in some cases (see Section 4.2.3.4).

Of the 12 apps that supported plaintext backups, only 4 provided a warning to users that exporting plaintext TOTP backups is risky. *2FAS*, *Authenticator Pro*, *andOTP*, and *Aegis Authenticator* (see Figure 4.1) each displayed a warning and required the user to verify their intent to export plaintext by clicking a checkbox, toggle, or confirmation button. *Authenticator Pro* displayed a detailed security warning before exporting plaintext to file, but also supported a feature that allowed the user to copy/paste a plaintext backup to the

clipboard with no warning.

### 4.2.3 Remote Backups *with* Encryption

More than half (15) of the apps we analyzed, comprising over 74 million installs, supported encrypted backups of TOTP data. However, the implementations of many of these apps introduced a range of vulnerabilities, up to and including giving the backup service the ability to decrypt the backup.

All of these apps implemented a similar architecture, in which they encrypted all or part of the TOTP backup using a symmetric encryption key and uploaded/exported the resulting ciphertext to a storage location, such as a third-party cloud service, Google Drive, or user-selected local and remote locations (see Table 4.1). We discuss the security and privacy impacts of the storage location in Sections 4.2.5 and 4.3.1.

All except one of the 15 apps derived the encryption key from a user-provided password. If an attacker obtains the backup, then they can try to crack the ciphertext just like they would a password hash. The feasibility of such offline attacks relies primarily on the strength of the password itself (discussed in Section 4.2.3.1) and secondarily on the KDF algorithm and how it is configured (discussed in Section 4.2.3.2).

*Microsoft Authenticator* was the only app that did not derive keys from user-provided passwords. Instead, it obtained randomly-generated AES-256 keys from a Microsoft key service. Using random keys is cryptographically ideal, but it introduces a significant key management challenge, which we discuss in Section 4.2.3.3.

The following subsections detail the data in Tables 4.2 and 4.3.

Abbreviated Name	Encrypted?	PII to use cloud backups					Backup Location	TOTP Data Leaked			Obtain Backup With...
		phone	email	name	dob	photo		secret	label	issuer	
Microsoft Authenticator	Yes*	Y	Y	Y	Y	-	activity.windows.com	Y	Y	Y	Microsoft account
Duo Mobile	Yes	-	Y	Y	-	Y	www.googleapis.com	-	Y^	Y^	Google account
Twilio Authy	Yes	Y	Y	-	-	-	api.authy.com	-	Y	Y	SMS only
Latch	No	-	Y	-	-	-	latch.elevenpaths.com	Y	Y	Y	Latch account
LastPass Authenticator	(Yes)	-	Y	-	-	-	(lastpass servers)	(Y)	(Y)	(Y)	Lastpass account
2FAS	No	-	Y	Y	-	Y	www.googleapis.com	Y	Y	Y	Google account
	Yes*	-	Y	Y	-	Y	www.googleapis.com	Y^	Y^	Y^	Google account
Yandex.Key	Yes*	Y	-	-	-	-	registrator.mobile.yandex.net	Y	Y	Y	SMS only
Salesforce Authenticator	Yes*	Y	-	-	-	-	authenticator-api.salesforce.com	Y	Y	Y	SMS only
TOTP Authenticator	Yes*	-	Y	Y	-	Y	www.googleapis.com	Y	Y	Y	Google account
Zoho OneAuth	Yes*	-	Y	-	-	-	accounts.zoho.com	Y	Y	Y	Zoho account
SAASPASS	No	Y	-	-	-	-	104.154.49.147	Y	Y	Y	SMS only

Table 4.2: Overview of the backup mechanisms that automatically sync data to the cloud. **Yes\*** indicates a serious security flaw in the implementation and/or usage of cryptography (see Section 4.2.3). **Y^** indicates the field is conditionally included in the backup as plaintext (see Section 4.2.5). Values in parentheses were obtained from documentation and observation only (see Section 4.3.4).

Abbreviated Name	Key Source	Password Min Len	KDF and Configuration	KDF Salt	Encryption Algorithm	Ciphertext Integrity	Decryption Heuristic
Microsoft Authenticator	Random*	n/a	n/a	n/a	AES-128-CBC	HMAC-SHA256	n/a
Zoho OneAuth	Password*	3	SHA-256 i = 1	none	AES-256-ECB	none	Base32
Salesforce Authenticator	Password*	4	PBKDF2-HMAC-SHA256 i = 10,000	random	AES-256-CBC	none	JSON
Yandex.Key	Password*	6	scrypt N = 2 <sup>15</sup> , r = 20, p = 1	random	Xsalsa20_Poly1305	AEAD	n/a
TOTP Authenticator	Password	8	SHA-256 i = 1	none	AES-256-CBC	none	JSON
Authenticator	Password	10	PKCS12-SHA256 i = 65,536	hard coded	AES-256-ECB	none	URI
App Authenticator	Password	10	PKCS12-SHA256 i = 65,536	hard coded	AES-256-ECB	none	URI
Auth0 Guardian	Password	1	(PBKDF2-HMAC-SHA1) (i = 10,000)	(random)	(AES-256)	(HMAC)	(n/a)
Authenticator Pro	Password	1	PBKDF2-HMAC-SHA1 i = 64,000	random	AES-256-CBC	none	JSON
2FAS	Password OpenPGP	1	PBKDF2-HMAC-SHA256 i = 10,000	random	AES-256-GCM	AEAD	n/a
Aegis Authenticator	Password	2	scrypt N = 2 <sup>15</sup> , r = 8, p = 1	random	AES-256-GCM	AEAD	n/a
andOTP	Password	4	PBKDF2-HMAC-SHA1 i = [140,000 - 160,000]	random	AES-256-GCM	AEAD	n/a
Twilio Authy	Password	6	PBKDF2-HMAC-SHA1 i = 10,000	random	AES-256-CBC	none	Base32
Duo Mobile	Password	10	argon2i m = 128 Mb, t = 6, p = 1	random	Xsalsa20_Poly1305	AEAD	n/a
LastPass Authenticator	Password	12	(PBKDF2-HMAC-SHA256) (i = 100,100)	(random)	(AES-256)	(HMAC)	(n/a)

Table 4.3: Cryptographic details of app backup mechanisms. The asterisk (\*) indicates that the app leaks the encryption key and/or password to the same service which stores the ciphertext, allowing that service to decrypt the TOTP backup (see Section 4.2.3.3). Square brackets indicate the min and max of a range, inclusive. Values in parentheses were obtained from documentation and observation only (see Section 4.3.4). The abbreviations for KDF configurations are: SHA/PKCS12/PBKDF2 (i = iterations), scrypt (N= CPU/memory cost, r = block size, p = parallelism), and Argon2 (m = memory, t = time/iterations, p = parallelism).

### 4.2.3.1 Password Policies

Most apps that derived encryption keys from user-provided passwords had severely inadequate password policies, making the encrypted backups vulnerable to trivial guessing techniques implemented in modern password cracking tools (RQ3). It is well established that people typically create weaker passwords on mobile devices [109, 71]. Best practices, such as rejecting weak passwords and nudging users to create stronger passwords, help to mitigate this risk.

The most important aspect of a password policy is the minimum length [49], but most apps accept incredibly short passwords; several apps even accept just a single character (see Table 4.3). A few apps did employ various levels of block lists. For example, *Duo Mobile* rejected known weak passwords and *Latch* used HaveIBeenPwned<sup>14</sup> to display a warning when the password was included in previous data breaches. It seems likely that many users would consider TOTP backups important, which suggests that password strength meters could nudge them to select stronger passwords [31, 107]; the only app to implement a strength meter was *Aegis*.

By default, *Auth0 Guardian* suggested a random password of length 10, but allowed a user to enter their own password instead. Similarly, *Authenticator* and *App Authenticator* could optionally suggest a password composed of 4 words selected from a list of 6,566 words ( $10^{15}$  permutations). Sadly, these apps also used a static salt, which largely negates any benefits of suggesting passwords (see Section 4.2.3.2).

### 4.2.3.2 KDFs and their Configurations

Most apps that we analyzed used a key derivation function (KDF) and/or KDF configuration that was wildly inadequate to meaningfully frustrate offline attackers (RQ3).

Among the apps that derive keys from passwords, *Zoho OneAuth* and *TOTP Authenticator* were at the highest risk to trivial offline attacks because they used a single round of SHA-256 as a KDF. SHA-256 is a prevalent cryptographic hash function that has been widely and heavily optimized to execute quickly in modern hardware. However, fast execution is the antithesis of the design goals for KDFs, which aim to execute slowly. SHA-256 is not a KDF and, therefore, should never be used alone for key derivation.

*Authenticator* and *App Authenticator* utilized the KDF defined in PKCS#12 [73], which is not appropriately hardened to mitigate offline attacks on modern hardware and has been deprecated in favor of PBKDF2 [72].

PBKDF2 [72] was the most commonly used KDF ( $n = 7$ ) among the apps we analyzed. Internally, PBKDF2 computes a Hash-based Message Authentication Code (HMAC) using a given cryptographic hash function. Of the 7 apps that used PBKDF2, 4 used SHA-1 and 3 used SHA-256. Most KDF configuration recommendations are made in the context of secure password storage, but they still serve as a useful reference point for key derivation. NIST [49] states that PBKDF2 iterations should be as many “as verification server performance will

---

<sup>14</sup><https://haveibeenpwned.com/>



allow,” but *at least* 10,000. While all of the apps that we looked at do 10,000 iterations or more, many argue this value is too low for modern usage. OWASP [79] highlights that values should take into account the underlying hash function and recommends 720,000 iterations for PBKDF2-HMAC-SHA1 and 310,000 iterations for PBKDF2-HMAC-SHA256. The highest PBKDF2 iteration count among apps we analyzed was only  $\sim 100,000$  (in *andOTP* and *LastPass Authenticator*).

Only 3 apps used modern memory-hard KDFs. While PBKDF2 is only CPU-hardened, modern KDFs are designed to also be memory-hardened, which significantly increases the cost of execution. *Yandex.Key* and *Aegis Authenticator* both used scrypt [80], while *Duo Mobile* used argon2i [11]. *Duo Mobile* used the *libsodium* library, which chooses different KDFs, configs, and algorithms depending on the available resources. The configuration for all of these apps exceeds the OWASP recommendations [79] for password storage. In Section 4.3.2, we argue that the backup mechanisms in TOTP apps should configure KDFs to run significantly more slowly.

While the vast majority of apps that we analyzed correctly utilized random salts when deriving keys from passwords, *Authenticator* and *App Authenticator* both used the hard-coded salt value “ROYALEWITHCHEESEROYALEWITHCHEESE,” which makes the TOTP backups generated by these apps vulnerable to rainbow table attacks [104]. Given the uniqueness of this value, we believe that *App Authenticator* is an unauthorized repackaged clone, since it is littered with ads while *Authenticator* is not.

### 4.2.3.3 Key Management

The encrypted TOTP backups created by several apps could be decrypted by the remote service storing the backup (RQ3). The security of a cryptographic architecture relies not only on how an encryption key is generated, but how that key is handled and stored. Several apps sent both the ciphertext **and** the encryption key (or password from which it was derived) to the same entity, allowing it to decrypt the TOTP backup.

Microsoft had the technical capability to decrypt the TOTP backups created by *Microsoft Authenticator* because it had access to both the encrypted TOTP backup and the associated key. As mentioned previously, *Microsoft Authenticator* obtained randomly-generated keys from a Microsoft key service instead of deriving keys from passwords. However, the app also stored the encrypted TOTP backup on a Microsoft controlled storage service. This behavior is clearly documented in a Microsoft blog post [111]. Microsoft engineers also acknowledged this design on Twitter,<sup>15</sup> but stated that the company implemented internal security procedures to reduce the risk of attack.<sup>16</sup> On the iOS version of the *Microsoft Authenticator* app, the ciphertext is stored in the user’s iCloud, removing Microsoft’s technical capability to decrypt the TOTP backup since it only has access to the key [111]. It seems that the

<sup>15</sup>[https://web.archive.org/web/20221003155439/https://twitter.com/Alex\\_T\\_Weinert/status/1195841144304758786](https://web.archive.org/web/20221003155439/https://twitter.com/Alex_T_Weinert/status/1195841144304758786)

<sup>16</sup>[https://web.archive.org/web/20221003155253/https://twitter.com/Alex\\_T\\_Weinert/status/1195841594814976000](https://web.archive.org/web/20221003155253/https://twitter.com/Alex_T_Weinert/status/1195841594814976000)

Android app could achieve a split-knowledge architecture like the iOS app by leveraging the Android Auto Backup system, or Google Drive directly, to store encrypted backups.

Several apps that derive keys from passwords also ran into the same vulnerability, but in less transparent ways. Each of *Yandex.Key*, *Zoho OneAuth*, and *Salesforce Authenticator* sent the backup password and the encrypted backup to domains controlled by each of those apps' developers, giving them the technical capability to decrypt the backups. The *Yandex.Key* app used Yandex servers to perform a password strength test, which should happen locally on the device instead. In its documentation, *Zoho OneAuth* states that “*The reason for [encrypting backups] is to make sure that your OTP secrets are stored securely and not accessed by anyone (including Zoho). You should note that only the encrypted secrets will be stored by Zoho and not the passphrase*” (bold theirs) [116]. It is unclear whether the bold text is meant to disclose that the password is sent to Zoho servers, but it certainly does not explain *why* it is transmitted. Regardless, there is no way for users to verify that Zoho does not, in fact, store the passphrase on the server. The *Zoho OneAuth* backup implementation seems to flagrantly violate the confidentiality goals defined in the documentation because Zoho has the technical capability to decrypt the TOTP backups. *Salesforce Authenticator* sent the password to Salesforce servers so that it could be used “to verify [users'] ownership of the backed-up accounts” [96] during recovery. Since *Salesforce Authenticator* also required an SMS OTP during authentication, the short recovery PIN does not provide enough additional security to warrant allowing Salesforce to decrypt the TOTP backup.

The *TOTP Authenticator* app suffers from a different key management issue: hard-coded keys. Backups uploaded to Google Drive are encrypted using a key derived from a static, hard-coded password, which is equivalent to hard-coding the key directly in the app source code. Anyone who decompiles the app can obtain the key, granting attackers with access to the backup the ability to instantly decrypt it. Surprisingly, the app does derive a key from a user provided password when encrypting TOTP backups exported to a file. The app should leverage the same behavior when backing up to Google Drive.

#### 4.2.3.4 How TOTP Backups are Encrypted

Of the 15 apps using encrypted backups, 5 used modern Authenticated Encryption with Associated Data (AEAD) primitives, while others used deterministic encryption (AES-ECB) and many did not provide integrity over the ciphertext (RQ3).

**AEADs** As seen in Table 4.3, 5 apps secured backups with an AEAD, primitives that encrypt and authenticate messages in one atomic API call, thus reducing the chance of developer error compared to employing encryption and Message Authentication Code (MAC) primitives individually. Xsalsa20\_Poly1305 was used by 2 apps (via the `libsodium` [56] `secret_box` API), while 3 others used AES-GCM. Next, we discuss the 8 apps<sup>17</sup> that used

---

<sup>17</sup>Some apps used AES with unknown modes of operation (see Table 4.3).

AES modes of operation that only provide confidentiality.

**AES-CBC** Of the 5 apps that use AES-CBC, 4 correctly generated a random initialization vector (IV), but the *TOTP Authenticator* app used an IV of all zeros. This flaw allows attackers to determine whether multiple backups start with the same plaintext blocks, but the real world impact is arguably immaterial. As mentioned in Section 4.2.3.2, this app uses a seriously flawed process to derive keys from user-provided passwords. Attackers will undoubtedly crack the password (and thus the key) directly, rather than analyze the ciphertext.

As mentioned in Section 4.2.2, the *2FAS* app does allow users to enable a feature that automatically encrypts and uploads TOTP backups to the user’s Google Drive. However, the app did not allow the user to enter a backup password *before* enabling the Google Drive backup mechanism, which resulted in existing TOTP data being uploaded to Google Drive in plaintext. Once a password was provided, all existing and future TOTP accounts were encrypted with AES-CBC using a random IV before they were backed up to Google Drive.

**AES-ECB** Alarmingly, 3 of the apps encrypted TOTP backups using AES-ECB, which is deterministic and does not provide indistinguishability under chosen plaintext attack (IND-CPA), commonly referred to as semantic security. Ironically, the use of AES-ECB in these particular apps is basically immaterial because they each have other serious security flaws that allow trivial decryption of the TOTP backups. *Zoho OneAuth* uses AES-ECB, but Zoho servers already have the technical capability to decrypt backups (see Section 4.2.3.3). The *Authenticator* and *App Authenticator* apps also use AES-ECB, but backups in these apps are already vulnerable to rainbow table attacks (see Section 4.2.3.2). Similar to the misuse of IVs in AES-CBC mode discussed previously, attackers are most likely to attack the backup passwords directly than hunt for clues in the ciphertext output by AES-ECB. Still, these apps should abandon AES-ECB and use an AEAD instead.

**Integrity** *Microsoft Authenticator* was the only app not to use an AEAD for encryption, while providing integrity over the ciphertext (using HMAC-SHA256). All other apps that used AES-CBC and AES-ECB had no cryptographic mechanism to authenticate the ciphertext before performing decryption. Instead, these 7 apps relied on a range of heuristics to determine whether the decryption succeeded. Checking whether the decrypted plaintext was a valid JSON object or URI were common techniques used by 5 apps. Interestingly, the *Twilio Authy* and *Zoho OneAuth* apps only encrypted the Base32 encoding of the TOTP secret. The random secret would normally prevent identifying whether the correct decryption key was used, but the Base32 encoding provides a reliable heuristic (see Appendix A.2). Offline attackers can use the same heuristic, so not including a MAC for CBC or ECB mode does not enhance security and seems to be an oversight.

**Asymmetric Encryption** *andOTP* was the only app that supported backups with asymmetric encryption. The app could send the plaintext backup to a third party app,<sup>18</sup> which encrypted the data using a PGP key and returned the resulting ciphertext. There is a clear risk of compromise when sending data to a third-party app. To reduce the attack surface, PGP functionality should be implemented directly within the app using trusted libraries.

#### 4.2.4 Android Auto Backup

Android 6.0 and above supports a backup system that automatically uploads an app’s data to the user’s Google Drive [9]. Android apps are opted into Android Auto Backup (AAB) by default, but developers can explicitly opt-out by setting `android:allowBackup="false"` in the app’s manifest file. The official docs [9] suggest that developers opt-out “...if [the] app deals with sensitive information that Android shouldn’t back up”. We argue that TOTP fields, especially the secret, are sensitive and should not be backed up via AAB without additional protections.

The AAB documentation [9] states that backup data uploaded to Google Drive is “...end-to-end encrypted on devices running Android 9 or higher using the device’s pin, pattern, or password.” We did not review the Android source code to verify implementation details, but it is apparent that encryption is used to protect app data. The use of end-to-end encryption is important to protect the backups at rest in Google’s data centers from internal attackers, but the device’s pin/pattern/password is likely to be low entropy and may not provide any meaningful protection against offline attacks.

Just over half (12) of the apps in our dataset explicitly opt out of AAB. We manually triggered the AAB functionality for each of the 10 apps that supported it and observed which data, if any, was restored after uninstalling and reinstalling the app. We could not get AAB to run without error for 3 apps that supported it.<sup>19</sup> While we could not observe the behavior for these apps, we included them in our analysis because AAB may work on Android versions that we did not test. *Google Authenticator* did support AAB, but did not backup any of the TOTP fields via this mechanism; it only backed up application settings, such as Dark Mode preferences.

Each of the remaining 6 apps whose AAB behavior we could observe backed up all of the TOTP fields (secret, issuer, and label) via AAB. The user’s Google account was a single point of failure for users of the 5 apps that relied solely on the security protections native to AAB. An attacker with access to the Google account could read the TOTP backup and generate valid OTPs for the user’s accounts.

The *Auth0 Guardian* app was the only app whose AAB behavior we could observe that added additional protections before the TOTP backups were sent to Google Drive via AAB. The app required the user to enter a backup password, which was used to enable the native encryption support in the Realm database [33] that the app used to store TOTP data and

---

<sup>18</sup><https://play.google.com/store/apps/details?id=org.sufficientlysecure.keychain>

<sup>19</sup>*Aegis Authenticator*, *andOTP*, and *FreeOTP Authenticator*

other app settings (see Table 4.3). We believe that the entire encrypted Realm database was backed up via AAB because the Auth0 app also required the backup password upon recovery.

It is also worth noting that even though we could not observe their AAB behavior, *andOTP* and *Aegis Authenticator* were the only apps that allowed the user to opt in/out of using AAB. Both apps required a backup password if AAB was enabled, which, we believe, would be used to derive a key and encrypt the TOTP backup before sending it to Google Drive.

### 4.2.5 Privacy Implications

This section explains how the backup mechanisms in TOTP apps can leak personal user information to third parties (RQ1).

**PII to Use App** Only 2 apps required users to provide personal information to the app developer in order to use the app at all, even if the backup mechanism was not enabled. *Twilio Authy* required the user to enter an email address and prove control of a phone number via SMS OTP or voice call. *Latch* required the user to create an account in order to use the app, which required an email address.

**PII to Enable Backups** Apps that supported local file exports and sharing did not request any personally identifiable information (PII) to use those features, but the use of plaintext exports (see Section 4.2.2) can have an obvious impact on users' privacy depending where they are sent and stored.

The ability to automatically sync backups to the cloud universally required users to divulge at least some PII so that they could be authenticated during recovery (see Table 4.2).

Of the 11 apps that supported a cloud sync backup mechanism, 8 required the user's email address and 5 required the user's phone number to utilize this feature. The *Microsoft Authenticator* app required a Microsoft account to enable cloud sync, which required the most PII of any app that we analyzed: phone number, email address, name, and date of birth. *Zoho OneAuth* also required account creation for cloud sync and collected user email address, country, and state.

The 3 apps that automatically sync backups to the user's Google Drive (i.e., separately from the aforementioned apps using AAB) used OAuth/OpenID to perform this upload on the user's behalf. Doing so granted each of those apps permission to read the user's primary email address, name, and account photo. Though not technically required for the backup functionality to work, the user's name and account photo fields are included in the narrowest set of permissions that developers can request.

**Leaking TOTP Labels and Issuers** Section 4.2.3 discussed app developers who have the technical capability to decrypt TOTP backups and read their content, including secret, issuer, and label. Here, we discuss TOTP data that is leaked directly to third parties in plaintext. Several apps that supported encrypted cloud sync only encrypted the TOTP secret and included all other TOTP fields in backups as plaintext.

Interestingly, *Duo Mobile* conditionally includes either the TOTP issuer or TOTP label in plaintext depending on whether the website/service is on an internal list of popular websites/services. If the website/service is a member of the list, then the TOTP issuer is included in plaintext, while the TOTP label is included in plaintext if the website/service is “custom” (i.e., not on the list). Both *Zoho OneAuth* and *Twilio Authy* only encrypted the secret, which meant the issuer and label are sent to the Zoho and Twilio servers in plaintext.

Functionally, there is no reason to avoid encrypting the TOTP issuer and label fields in the TOTP backups and it is not immediately obvious how this decision improves the user experience. These plaintext fields appear to have no impact whatsoever on the UX of the *Zoho OneAuth* app; no TOTP data is displayed during recovery unless the correct backup password is entered. In the *Duo Mobile* and *Twilio Authy* apps, however, the TOTP issuer and label are displayed in the app even if the backup password is not entered at all, or is incorrect. At best, one could learn which accounts they may be locked out of if they cannot recall their backup password. While backups generated by *Duo Mobile* are stored on Google Drive, the *Twilio Authy* app and *Zoho OneAuth* app each store backups on their own servers. This means that any user of *Twilio Authy* or *Zoho OneAuth* who enables cloud backups is unknowingly sending those companies the names of the websites/services they use and the usernames for their accounts on those platforms.

## 4.2.6 Reliance on Passwords, SMS, and Email

Of the 19 apps that supported automatically uploading TOTP backups to the cloud (via cloud sync features and/or AAB), 15 required the user to authenticate to the cloud service storing the backup, while 4 relied solely on SMS OTP to authenticate users during recovery (see Tables 4.1 and 4.2) (RQ2).

The *SAASPASS* app supported a dangerous combination of uploading plaintext TOTP backups to the cloud and relying solely on SMS OTP to authenticate users during recovery. Any attacker who leverages any of the well-known and numerous techniques to hijack the user’s phone number will gain immediate access to the full TOTP backup, including the TOTP secrets. The app does allow users to optionally enforce a 20-hour delay on sending the recovery OTP via SMS, which can give the user critical time to switch to a different TOTP app while rotating their TOTP secrets and/or regain control of their phone number. It is unclear how many users discover this option in the security menu and enable it in practice.

By default, each of *Twilio Authy*, *Yandex.Key*, and *Salesforce Authenticator* also relied solely on SMS OTP to authenticate users during recovery, but did encrypt TOTP backups using a key derived from a password before uploading them to the cloud. To compromise

the backup, an attacker who hijacks the phone number will still need to conduct an offline attack to guess the backup password. Presumably, many users will realize that their phone number has been compromised once their phone stops working [18] and take action. Thus begins a race. In addition to regaining control of their phone number, the user should begin rotating their secrets on each individual account protected by TOTP. The question is whether they can rotate everything before the attacker successfully cracks the TOTP backup, enabling them to generate valid OTPs and attempt to log into their accounts. If the backup password could be cracked quickly, then chances are high that it was relatively weak, which is unsurprising considering the password policy issues discussed in Section 4.2.3.1. Given the fact that password reuse is rampant [83], it seems likely that the user may also have weak account passwords, which may allow the attacker who hijacked the user’s phone number to fully compromise their online accounts protected by TOTP 2FA.

The remaining 15 apps that supported remote cloud backups required the user to log into to an account on the cloud service to obtain the TOTP backup. This begs the question: *what are the authentication mechanisms for those accounts?*

To use the *Latch* app at all, we were required to create an account on the Latch website, which only required a username and password. There were no indications of any support for 2FA. During recovery, the backup could be obtained with only the username and password.

Cloud backups on *Microsoft Authenticator* required creating a Microsoft account. In addition to a username and password, creating this account required proving control of a phone number and providing an email address. To obtain the backup during recovery, we were required to prove control of the phone number again (i.e., SMS 2FA).

*LastPass Authenticator* actually relied on the LastPass Password Manager app, developed by the same company, to encrypt and store TOTP backups. The password manager is free to use, but requires creating an account by providing an email and password. The *LastPass Authenticator* app then requires that at least one 2FA option is enabled on the password manager account. However, the only 2FA options freely available on LastPass accounts that did not rely on possession of a device was printable recovery codes. Of course, if the user enables TOTP 2FA or Push 2FA and only has a single device, then they will not be able to log into their LastPass account in the event that they lose their phone.

A Zoho account was required to enable cloud backups in *Zoho OneAuth*, which required an email and password. Zoho does support multiple methods of 2FA [57], including SMS 2FA, but none were required.

As mentioned previously, the user’s Google account is commonly used to store remote backups; apps upload TOTP backups to Google Drive directly, or via Android Auto Backup. In an effort to combat low adoption rates and increase the security of accounts, Google announced in 2021 that it would begin requiring hundreds of millions of users to enable 2FA on their accounts [70]. Arguably, this decision reflects a larger, industry-wide paradigm shift [7] and unquestionably improves the overall security posture of account security. However, at the time of our analysis, the workflow to enable 2FA on a Google account required users to initially choose one of three specific methods: Push 2FA, SMS 2FA, or a security key. Google does support enabling several additional 2FA methods after initial setup [1], but it

seems almost certain that the vast majority of users will stick with either SMS 2FA or Push 2FA given the general lack of knowledge about 2FA and its security benefits [58, 88, 22], low organic adoption rates [44], that adoption of security keys is quite rare among the general population [38], and that these alternative 2FA methods are completely optional. If SMS 2FA is enabled, then the Google account is protected by the same exact technologies that users of TOTP are likely trying to avoid: passwords and SMS. If Push 2FA is enabled, then the user could face account lockout across all of their online accounts protected by TOTP 2FA if their device is inaccessible; they will be unable to generate OTPs because the device on which the TOTP secrets were stored is lost and, at the same time, they will be unable to recover those TOTP secrets because they were backed up to their Google account, which requires them to approve a notification sent to the lost device to log in. Registering multiple personal devices for Push 2FA can help recovery if only a single device is lost, but many users only have a single device.

For cloud-based backup mechanisms, this account recovery conundrum inevitably reduces the security of the TOTP 2FA scheme to just another layer of the same authentication mechanisms that TOTP 2FA is meant to supersede: username/password, SMS, and/or email (RQ2).

## 4.3 Discussion

In this section, we discuss the dangers of plaintext backups, make recommendations, describe our responsible disclosure, and discuss the limitations of this study.

### 4.3.1 The Dangers of Plaintext Backups

The dangers of plaintext backups are largely self-evident, but warrant some unpacking to gauge the real-world risk.

Universally among the apps we analyzed, an attacker with access to the plaintext TOTP secret also learned the names of the websites/services on which the user had accounts and/or the usernames for those accounts, allowing them to leverage classic password attacks. Plaintext TOTP backups grossly undermine the security benefits of TOTP 2FA.

The majority of apps that support plaintext backups do so via file export or sharing. File exports from a TOTP app will not help a user recover if the export is stored locally on the lost device. Sending plaintext backups remotely can leak the TOTP data to every actor involved in transporting and storing the backup. For example, if a user sends the plaintext file via email, then it is stored in the sender's outbox and the recipient's inbox. Some users may understand these risks and choose a secure alternative, such as sending the backup to another personal device via an end-to-end encrypted chat app, such as Signal.<sup>20</sup> However, we expect that many users will not grasp the sensitivity of plaintext backups since so few apps warned users about the associated risks (see Section 4.2.2).

---

<sup>20</sup><https://signal.org/>



One use case in which plaintext exports are particularly useful is migration between apps. Many apps supported importing the plaintext backups from other TOTP apps. As long as the plaintext export is deleted after the local migration is complete, there should be little to no risk of compromise.

Google Drive is a common storage location for TOTP backups; many apps store backups there via Android Auto Backup, several apps upload there directly using SDKs, and many Android users will likely choose it as a storage location via the sharing menu. Storing plaintext TOTP backups in Google Drive simply outsources responsibility for securing them to Google. Many users may, in fact, have a threat model in which this is a perfectly reasonable backup solution given Google’s existing security mechanisms. Google Drive is also operated by a separate company than all of the apps that use it for storage, potentially allowing the TOTP backups to hide among the masses. We argue that these assumptions do not hold for *Latch* and *SAASPASS* because their employees know that their servers are storing plaintext TOTP backups, which increases the risk of internal attacks. The security of remotely-stored plaintext backups against external attackers is directly dependent on the authentication mechanisms protecting that account. An attacker who compromises a user’s *Latch* or Google accounts can simply install the relevant TOTP app to automatically restore **all** of the user’s TOTP secrets.

### 4.3.2 Recommendations

According to the Google Play Store, the TOTP apps that we analyzed have a collective install count of over 180 million (see Table 4.1). The following recommendations will help app developers improve their backup and recovery mechanisms to address security vulnerabilities and respect user privacy.

Apps should consider not supporting plaintext backups. If they are supported, then users should be clearly warned about the associated risks.

Including any TOTP fields in backups as plaintext violates user privacy. Apps should encrypt all TOTP fields, including the secret, issuer, and label. Apps that rely on remote key servers to generate/store random keys should choose different entities for storage of keys and ciphertext; if both are stored with a single entity, then they can decrypt the TOTP backups.

We have several recommendations for apps that derive keys from passwords. First, they should implement well-established best practices to encourage users to create strong passwords (see Section 4.2.3.1). Second, **NEVER** allow the backup password to leave the app. Once the key is derived from the password, the password should be wiped from memory and the key should be securely stored on the device using the Android Key Store <sup>21</sup> so that it can be used to encrypt TOTP accounts added in the future.

Finally, we recommend that TOTP apps that derive keys from passwords should capitalize on the fact that the KDF operation happens so infrequently and configure it to run

---

<sup>21</sup><https://developer.android.com/training/articles/keystore>

significantly slower than existing recommendations for password storage. In contrast to account authentication and unlocking password managers, which can easily happen many times per day, TOTP apps should only ever perform key derivation two times: when the user first enables the backup mechanism, and when the user is attempting to recover. In fact, this is the exact architecture that several apps already implement (e.g., *Duo Mobile*). The original script paper [80] asserted that 5 seconds is a reasonable amount of time to wait for file encryption. While that value is subjective, Egelman et al. [32] did find that people are willing to endure longer delays in security contexts when they are informed of the threat model and how the delay enhances their security. Given that TOTP apps already show a 30 second countdown<sup>22</sup> indicating when the OTP will be regenerated, we propose that TOTP apps should reuse this familiar UX and dynamically calculate the KDF configuration such that it takes 30 seconds to execute. While the KDF executes, the visual countdown should be displayed along with an explanation of how the delay increases defense against offline attacks and that it will only occur again during recovery. It is likely that the willingness of users to wait has an upper bound regardless of explanation. We believe 30 seconds is a reasonable starting point, but future work should explore this limit empirically.

TOTP apps that derive keys from passwords should adopt the Argon2 KDF because it allows developers to independently configure memory and CPU parameters. As a result, all devices should be able to extrapolate from a small set of tests a KDF configuration that should execute within the desired clock time. Encouragingly, the *andOTP* app did dynamically calculate the number of rounds of PBKDF2 required to keep the clock time within 1 second, but only used this technique when encrypting the TOTP backup in Android Auto Backup (see Section 4.2.4) and not when backing up to Google Drive. Higher end devices could consider the historical trends in device specs (e.g., the median amount of memory of phones sold within the last 2 years) to ensure that recovery can run without error on lower end devices, even if it takes longer than 30 seconds. For example, a user who loses their brand new phone would reasonably expect to be able to recover on an older device running the same TOTP app.

### 4.3.3 Responsible Disclosure

Here, we outline our best efforts to disclose substantive issues to the respective app developers and summarize the responses we received as of November 4, 2022.<sup>23</sup> We felt there was nothing to disclose for the following 6 apps: *Google Authenticator*, *LastPass Authenticator*, *FreeOTP Authenticator*, *Authenticator Pro*, *Aegis Authenticator*, and *Auth0 Guardian*. The developer of *andOTP* announced its deprecation [74] after our analysis, so no report was filed. *TOTP Authenticator* did not respond to our email and Twitter communications asking for a security contact, so no report was filed. We contacted each of the remaining 14 app developers and gave them at least 90 days to respond to our report.

---

<sup>22</sup>While most TOTP apps are capable of using any time window defined by the server, 30 seconds is almost universally used in practice.

<sup>23</sup>See Section 4.4 for supplemental material available online.

The issues in the *Twilio Authy* app that we disclosed to Twilio in 2020 [46] (v24.3.1) persisted in the app over 2 years later (v24.8.5). Twilio stated in October 2022 that they are “committing to increase the length of the Backup password” and “significantly increasing the number of [PBKDF2] iterations.” In response to our disclosure that the Authy backup mechanism sends the TOTP issuer (i.e., website/service name) and label (i.e., the account username) to Twilio servers in plaintext, Twilio stated that users are able to set the label field to any custom value and that these fields are required “...as an aid to help the users to know what tokens they encrypted in multi-device scenarios.” We find it extremely unlikely that many users change the default value of the TOTP issuer, which is almost universally set as their username by the website on which they are enabling TOTP, and wanted to know the percentage of users who take this action in practice. However, Twilio claimed that they do not track changes to the TOTP issuer field. Twilio stated that they are considering updates to their privacy policy. Twilio uses BugCrowd, but we chose not to disclose via that platform because “Twilio does not permit public disclosure at this point in time.”<sup>24</sup>

*Latch* did not rule out our suggestion of allowing users to opt in/out of TOTP backups, but rejected our strong recommendation to implement end-to-end encryption, stating that the current behavior of sending TOTP backups to Latch servers in plaintext was “by design” and “intended.”

Microsoft confirmed the backup mechanism in their Android app was “by design” and highlighted their internal security mechanisms. They did not respond to our suggestion to store ciphertext in Google Drive instead of a Microsoft storage service, nor our inquiry about why a more robust backup mechanism was implemented on the iOS version of the app.

Duo confirmed our report and pointed us to the Duo Privacy Data Sheet[28], which they said listed Google as a sub-processor and disclosed the collection of username/email. Regarding their backup design, Duo stated, “[b]y allowing users to see their accounts in a non-restored state, Duo’s goal is to help facilitate them setting up their accounts with their services (e.g. Amazon) more easily. They do not have to recall every service they set up OTP accounts for on their own.”

Salesforce responded to our disclosure report by linking to two support articles [96, 97] on backup and recovery. Each of these articles stated “...your encrypted TOTP data is stored on Salesforce servers...During backup and restore events, your passcode is used to verify your ownership of the backed-up accounts.” Neither the documentation nor the response from Salesforce addressed the fact that Salesforce has the technical capability to decrypt TOTP backups (see Section 4.2.3.3).

The developer of *Authenticator* was very receptive to our disclosure and released an updated version of the app (v1.2.4) that (1) switched from static to random salts; (2) switched from AES-ECB to AES-CBC; (3) set minimum password length to 20; and (4) warned of plaintext export risks.

The developer of *Code Generator* said they would update the app when they had “...enough available time and resources...” Communication with *2FAS* ceased after a developer asked

---

<sup>24</sup><https://bugcrowd.com/twilio>

for our PGP key, which we provided.

At the time of publication, the remaining 6 app developers to whom we disclosed our findings never replied. Discussions are on-going with several companies.

#### 4.3.4 Limitations

We focused on Android only, so future work should analyze the behavior of TOTP 2FA apps on iOS. We expect that most apps will exhibit the same behavior on both platforms, but we know this is not the case for all apps (e.g., *Microsoft Authenticator* discussed in Section 4.2.3.3).

As we discuss in Section 4.2.4, we could not get Android Auto Backup to run without error for several apps.

We were unable to verify the cryptographic primitives used by the *LastPass Authenticator* and *Auth0 Guardian* apps. The *LastPass Authenticator* app actually relies on the LastPass Password Manager app<sup>25</sup> (developed by the same company) to enable encrypted cloud sync for TOTP backups. We found documentation that it derives keys from the master password using 100,100 rounds of PBKDF2 with random salts, calculates an authentication hash, and encrypts data using AES-256 [76, 53]. We observed in the decompiled code that *Auth0 Guardian* uses RealmDB to store application data, which has native support for encrypting data using AES-256 for confidentiality and an HMAC for integrity[33]. As appropriate, the value for these two apps in Tables 4.1, 4.2, and 4.3 are clearly marked as “from documentation and observation only” where we were not able to verify the claims.

### 4.4 Supplementary Materials

We strove to make our work fully verifiable and reproducible. To that end, there are numerous supplementary materials available online at <https://allthingsauth.com/totp-apps>.

---

<sup>25</sup><https://play.google.com/store/apps/details/?id=com.lastpass.lpandroid>

## Chapter 5

# Exploring Account Lockout Among TOTP Users

Chapter 4 outlined many security and privacy issues that exist within the backup mechanisms implemented in popular TOTP apps and highlighted their potential impact on TOTP users. I wanted to learn more about the real-world impact of these technical findings. For example, how often do users of TOTP apps actually utilize these available backup mechanisms? If they are rarely used, do the security issues matter less? How do people feel about the personal information that TOTP apps have been seen collecting? If they are comfortable with this data collection, would it decrease the urgency to change how these apps function?

This chapter presents unpublished work<sup>1</sup> that explores the real-world experiences of people who protect their personal, non-work accounts with TOTP and how they plan for account recovery to prevent permanent account lockout.

We focused on consumers using TOTP on personal, non-work accounts because people who use TOTP for work accounts often have more recovery options and less choice about which TOTP app to use and how it is configured.

First, employees who lose access to their TOTP app could likely contact some internal support channel to reset their credentials, such as IT or a help desk. This type of credential reset is only possible because the work organization knows who the employee actually is and can verify their identity. Consumers, on the other hand, are often anonymous to most of the online services they use. If a user loses access to their TOTP app and cannot generate the OTPs required to login to their account, it is difficult (if not impossible) for consumer services to differentiate between the legitimate account holder and an attacker.

Second, employees using TOTP for work accounts may not have a choice in which TOTP app they use and how it is configured. Internal policies within an organization often dictate which software and apps employees are required to use, especially for security purposes. Some also dictate how those software and apps must be configured on work devices. Mobile device management software, for example, could force cloud-backups to be enabled within a

---

<sup>1</sup>Authors: Conor Gilseman, Diana Kohr, and Serge Egelman.

TOTP app.

In the context of protecting personal, non-work accounts with TOTP, our research questions included:

**RQ1** How do users perceive the risks of account lockout?

**RQ2** What precautions do users take to reduce the risk of account lockout?

**RQ3** What privacy concerns do users have about TOTP apps?

## 5.1 Methods

We conducted two separate surveys for this study: (1) a brief screening survey<sup>2</sup> to identify individuals who had used specific TOTP apps; and (2) a main survey<sup>3</sup> targeting users of those TOTP 2FA apps to gather information about their experiences.

We iteratively pilot tested each survey to evaluate question comprehension, uncover unforeseen issues, and verify timing. Each survey was reviewed by UC Berkeley IRB. We deliberately configured each Qualtrics survey to **not** collect any personally identifiable information (PII): IP addresses or geolocation information were not collected. We recruited participants on Prolific<sup>4</sup> who were 18 or older, lived in the US, and spoke English. We compensated approved submissions at a rate of at least \$15/hour, prorated per minute.

The survey contained logic to identify when submissions would be rejected (e.g., failed an attention check, failed comprehension quiz, etc.) and automatically redirected participants back to Prolific. In these cases, each participant was informed why their submission would be rejected and asked to return their submission without penalty. Prolific users are penalized each time a researcher rejects their submission, so we waited at least 48 hours before manually rejecting submissions.

### 5.1.1 Screening Survey

We designed the screening survey to identify people who had secured their personal, non-work accounts with one or more of the top four TOTP apps identified in Chapter 4: Google Authenticator, Microsoft Authenticator, Authy, and Duo Mobile.

We explicitly instructed participants to consider only *personal, non-work accounts* when reporting their 2FA usage<sup>5</sup>. We made this decision because employees typically have organizational support, such as IT help desks, to assist with account recovery, thereby bypassing the built-in recovery mechanisms of TOTP apps. Moreover, internal policies often dictate

---

<sup>2</sup>The screening survey ran from November 24, 2024 - February 13, 2025.

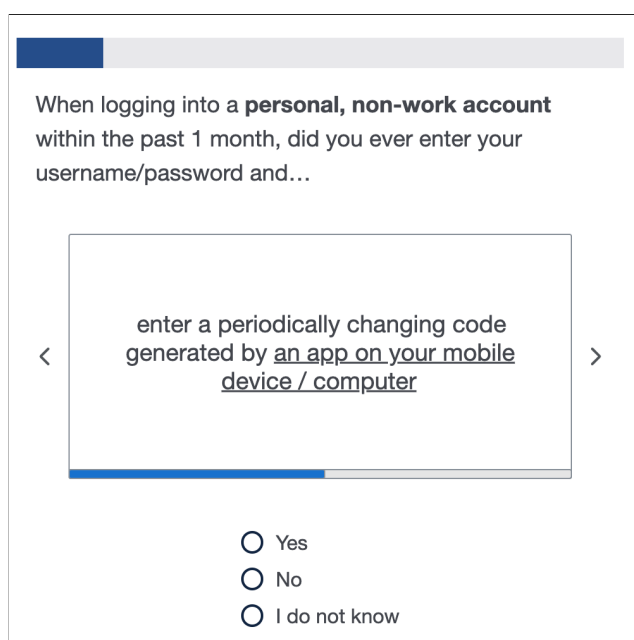
<sup>3</sup>The main survey ran from March 14, 2025 - April 14, 2025.

<sup>4</sup>[prolific.com](https://prolific.com)

<sup>5</sup>For brevity, the rest of this paper uses the term *accounts* to mean personal, non-work accounts.

which apps employees must use and how to configure them. For example, Florencio and Herley [40] observed that users created stronger passwords when alternatives were unavailable, such as for government or university websites.

Previous research has shown that most people do not know what 2FA actually is [108], so asking questions with technical acronyms like “Do you use 2FA?” will not produce meaningful data. Instead, the screening survey asked them to choose from a predefined list of the actual actions that they would take when using a given 2FA method (see TOTP description in Figure 5.1). We believed that this question design would be easier for participants to understand, provide more accurate data on which methods of 2FA they actually use, and avoid user confusion when a single app (e.g., Microsoft Authenticator and Duo Mobile) implement multiple different methods of 2FA (e.g., TOTP and Push notifications). The action descriptions underwent several design iterations in an attempt to make them as easy to understand as possible.

The image shows a survey prompt within a rectangular frame. At the top, there is a horizontal bar with a blue segment on the left and a light gray segment on the right. Below this bar, the text reads: "When logging into a **personal, non-work account** within the past 1 month, did you ever enter your username/password and...". In the center, there is a smaller rectangular box with a light gray border. Inside this box, the text says: "enter a periodically changing code generated by an app on your mobile device / computer". To the left and right of this central box are gray chevron symbols "<" and ">". Below the central box, there is another horizontal bar, similar to the one at the top, with a blue segment on the left and a light gray segment on the right. At the bottom of the frame, there are three radio button options: "Yes", "No", and "I do not know".

When logging into a **personal, non-work account** within the past 1 month, did you ever enter your username/password and...

< enter a periodically changing code generated by an app on your mobile device / computer >

☐ Yes  
☐ No  
☐ I do not know

Figure 5.1: Prompt asking about which 2FA options a participant had used and the description for the TOTP option.

The survey informed participants that qualifying responses might lead to a follow-up survey with additional compensation, so the survey asked about all of the prominent methods of 2FA (see Appendix B.1) to avoid social desirability bias leading people to say they used TOTP when, in fact, they had not.

Participants who said they had taken the action associated with TOTP 2FA received a follow up question asking whether they had used each of the 4 TOTP apps included in the main survey. Each TOTP app was listed by name and included a screenshot of the app

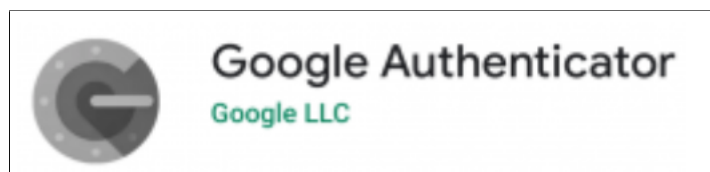


Figure 5.2: Example screenshot shown to participants when they were asked which TOTP apps they had used.

name, app icon, and developer name from the Google Play store (see Figure 5.2). For each app that the participant currently or previously used, they were asked to confirm on which platform(s) the app was used (e.g., Android or iOS) and that the app was used for personal, non-work account(s) as opposed to just work accounts.

We invited participants who used at least one TOTP app for personal, non-work accounts on Android or iOS to complete the main survey.

### 5.1.2 Main Survey

The beginning of the survey presented participants with two simple comprehension questions that served both as attention checks and as background information. We explained that the “periodically changing code” used during login (see Figure 5.1) is called a “TOTP code,” that it typically refreshes every 30 seconds, and that we would refer to apps generating such codes as TOTP apps. We also showed them an image of a sample TOTP app displaying OTPs for fake accounts. We excluded any participant who failed the comprehension quiz twice.

We categorized participants into two groups: (1) “current users,” who currently used one of the 4 TOTP apps listed in the survey; and (2) “previous users,” who had used at least one of the 4 apps in the past, but no longer did. These two group names appear throughout this dissertation. The following sections detail the questions we posed to all participants and those shown only to current users.

#### 5.1.2.1 Questions for all users

After the comprehension quiz, the survey assessed participants’ basic understanding of TOTP-related lockout risk. We presented the following vignette:

Alice can login to her account with just a username/password. The site has no customer support, but does allow users to reset their password using a ‘Forgot password’ link. Alice takes the following steps to enable TOTP on her account:



- 1) Installs a TOTP app that can only add new accounts and display TOTP codes
- 2) Scans the setup QR code displayed in the browser with the TOTP app
- 3) Types a TOTP code into the browser to complete the setup process

If Alice loses access to her TOTP app, will she be able to generate the TOTP codes required to login to her account?

We assert that the only correct answer is *no*. Initially, Alice could log into her account with just a password, so there was no 2FA method enabled. Alice set up TOTP 2FA on one device, did not back up recovery codes nor screenshot the setup QR code, and the app lacked any backup feature.

We asked all participants whether they had experienced account lockout after failing to enter a TOTP code within 5 minutes. We initially asked the question in binary yes/no format, but pilot tests showed very few “yes” responses. To encourage reflection, we introduced the 5-minute criterion and treated any answer other than “never” as a possible lockout event. We followed up with questions about the reason for the lockout and treated all reasons other than one that can be summarized as “I was being lazy” as an actual case of account lockout. If participants had experienced lockout, we asked them how they attempted to regain access to their accounts, how long before they regained access, and whether they made any changes to their account login settings as a result of this experience.

As we found in Chapter 4, various TOTP apps failed to appropriately protect the website name, account username, and TOTP secret while storing backups in the cloud and/or collected these fields in plaintext. Therefore, we asked participants which fields they expected the app developer or other third-parties could read and how they would feel if those fields were collected.

### 5.1.2.2 Questions for current users

This section details how we asked current users about their experiences with 5 specific backup strategies that were directly relevant to people who protect their accounts with TOTP. The survey avoided asking participants to rely on their memory as much as possible, so it did not ask previous users questions about the backup strategies they followed for the TOTP apps they used in the past.

First, we wanted to learn whether participants were aware of the backup mechanisms that were supported in their app: (1) cloud-based and (2) QR-based backups. Second, we wanted to learn whether users followed other backup strategies that were not directly related to their app: (3) multiple devices, (4) account recovery codes, (5) screenshots of setup QR-codes.

**Cloud-based and QR-based backups** Cloud-based backups are the most common backup mechanism supported in popular TOTP apps (see Chapter 4) and all of the apps in the survey also supported this feature.

We first asked participants to predict whether their app supported cloud-based<sup>6</sup> or QR-based<sup>7</sup> backups without consulting the app. We then informed them that their app did support cloud backups (and QR backups for Google Authenticator users), and asked whether they believed the feature was enabled.

One important goal of the survey was to gather data about which backup strategies are followed by people who use TOTP in practice. To this end, the survey asked people to open the TOTP app on their device and check whether cloud-based backups were actually enabled or disabled. To increase the quality of responses, participants were given detailed instructions on how to check the status of this feature, including written instructions and annotated screenshots specific to each TOTP app for both iOS and Android (see Appendix B.3). Additionally, users were informed that the next question in the survey would be an attention check that asked for a word documented in the linked instructions and that there would be no back button. Users who did not provide the correct attention word were excluded from analysis. Based on their response, the survey asked an open-ended question about why they had chosen to enable/disable the feature and several follow up multiple choice questions about their experiences with the feature.

**Multiple devices** The survey asked participants whether they had installed the app on multiple devices. If a user has the app setup on multiple devices, then they could retrieve a TOTP code from a secondary device if a primary device is lost. There are several ways that a user could get TOTP codes on multiple devices. For example, they could scan the setup QR-code with multiple different devices while enabling TOTP on their accounts, or they could enable cloud-based backups, which often also synchronize TOTP codes across linked devices.

**Account recovery codes** Websites that support TOTP will often prompt the user to download account recovery codes while enabling TOTP on their account (see Figure 5.3). These codes are frequently the last resort for logging into personal, non-work accounts and without them, many sites inform users that they could face account lockout. We asked current users whether they had saved account recovery codes for some or all of their accounts, where they saved them, and how confident they were that they could access the recovery codes.

---

<sup>6</sup>Described as “The app can backup your accounts to a remote storage service.”

<sup>7</sup>Described as “The app can display a QR code that allows you to manually transfer all of your accounts to another device.”

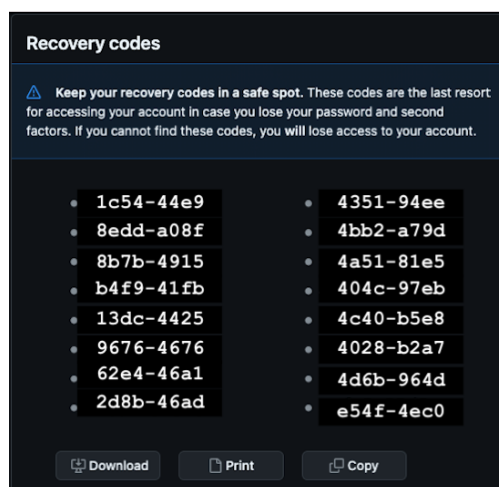


Figure 5.3: Example of account recovery codes often displayed when a user is enabling TOTP on their account.

**Setup QR-codes** Almost all websites that support TOTP display a QR-code to the user during the setup process (see Figure 5.4). As detailed in Chapter 2, this setup QR-code contains a static secret, which is used by the TOTP app to generate valid TOTP codes for the user’s account. Since, unlike the generated TOTP codes, this TOTP secret does not change over time, a user could save a screenshot of the setup QR-code somewhere safe as a backup mechanism. In the case where they lost their device(s), they could simply install the TOTP app on a new device and scan the setup QR-code again to regain the ability to generate valid TOTP codes for their account. We asked current users whether they had saved any screenshots of setup-QR codes, where the screenshots were stored, and how confident they were that they could access the screenshots.

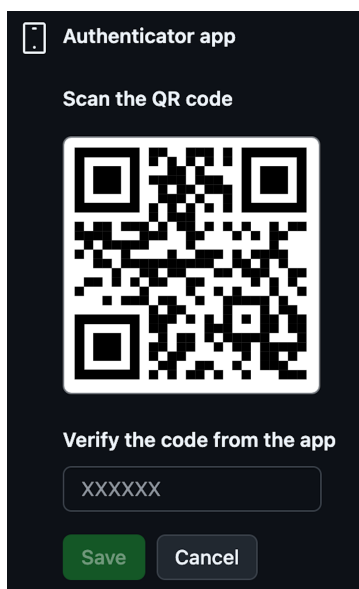


Figure 5.4: Example of a setup QR-code displayed when a user is enabling TOTP on their account.

### 5.1.2.3 Qualitative coding of open-ended responses

The main survey contained several open-ended questions that allowed participants to respond in their own words. The first two authors qualitatively analyzed user responses by collaboratively developing a code book for each question and individually applying each code book to all corresponding responses. The Kupper-Hafner concordance statistic [62] was used to calculate the inter-rater reliability (IRR) score for each question because multiple labels could be applied to each response.<sup>8</sup>

### 5.1.2.4 Post-hoc statistical testing

The goal of this work was to explore the real world experiences of people who protect their personal, non-work accounts with TOTP and how they plan for account recovery to prevent permanent account lockout. Given the exploratory nature of the survey and its analysis, we did not formulate any null hypotheses. The statistical tests presented in this paper have not applied any correction factor to account for post-hoc testing. We present the statistical tests simply to highlight interesting relationships that warrant further study in the future.

---

<sup>8</sup>We refer to this statistic as “Kupper-Hafner IRR” throughout the paper.

## 5.2 Results

### 5.2.1 Participant Population

This section documents how many participants were invited to each survey, how many responses were rejected, and how many were analyzed.

#### 5.2.1.1 Screening survey population

The screening survey was designed to identify participants who used at least one of the TOTP apps for personal, non-work account(s) on Android or iOS. We had very high recruitment numbers for the screening survey because pilot testing confirmed our intuition that identifying this narrow subset of people would be difficult.

In total, 5327 submissions were made by 5298 unique people. Although the Qualtrics survey was configured to allow only one submission per person, various issues led to 29 participants making multiple submissions ( $n = 58$ ). We reviewed the first complete submission, if any, for each participant, which yielded 5298 total submissions.

We rejected 908 (17%) submissions from our analysis. Of those, 515 (57%) were voluntarily returned or manually rejected on Prolific for: not consenting to participate (2), failing the attention check within 2FA selection (308, 60%), failing to complete the survey (44, 9%), and having seemingly conflicting answers (161, 31%). We also rejected participants from our analysis if they chose a fake TOTP app during TOTP app selection (395, 7%).

Answers seemed in conflict when participants claimed to have taken the action associated with TOTP 2FA, did not report using any of the TOTP apps that *were* mentioned in the survey, and also responded “No” when asked whether they had used any other apps or software to generate TOTP codes that *were not* mentioned in the survey. We do not see how someone could have both been required to enter a TOTP code generated from a piece of software and also appear not to have ever actually used any such software. We messaged all of these participants on Prolific to ask for an explanation, but few responded and none had a reasonable explanation, so these responses were rejected from our analysis.

Ultimately, 4775 (90%) submissions were approved and compensated on Prolific with a median completion time of 1.78 minutes. Our final analysis included 4380 (83%) of screening survey submissions.

Of the submissions we analyzed, 2070 (47%) participants reported using TOTP. Of those TOTP users, 750 (36%) were eligible to take the main survey because they used at least one of the TOTP apps for personal, non-work account(s) on Android or iOS. The invite rate for all participants who took the screening survey was 17%.

#### 5.2.1.2 Main survey population

Although participants were only allowed to make a single submission, various issues led to 22 participants making multiple submissions ( $n = 48$ ). We reviewed the first complete submission, if any, for each participant, which yielded 439 total submissions.

We rejected 107 (24%) submissions from our analysis. Of those, 73 (17%) were voluntarily returned or manually rejected on Prolific for: failing the comprehension quiz twice (10, 2%), not selecting any of the TOTP apps listed in the survey (3, 1%),<sup>9</sup> only using TOTP app(s) for work purposes (17, 4%),<sup>9</sup> failing two or more attention checks (2, 0%),<sup>10</sup> and failing to complete the survey (41, 9%). We also rejected participants from our analysis if they failed just a single attention check (35, 8%), even though they had to be compensated on Prolific.<sup>10</sup> Ultimately, 330 (76%) submissions from the main survey were included in our analysis.

The median completion time for all users compensated on Prolific was 9.6 minutes. Current users had a median completion time of 10.83 minutes while previous users had a median completion time of 4.47 minutes.

The 330 total participants included 293 (89%) current users and 37 (11%) previous users. The collected demographics for all analyzed submissions to the main survey are detailed in Table 5.1. Figure 5.5 shows the breakdown of how many current users were most familiar with each app and on which platform the app was used.

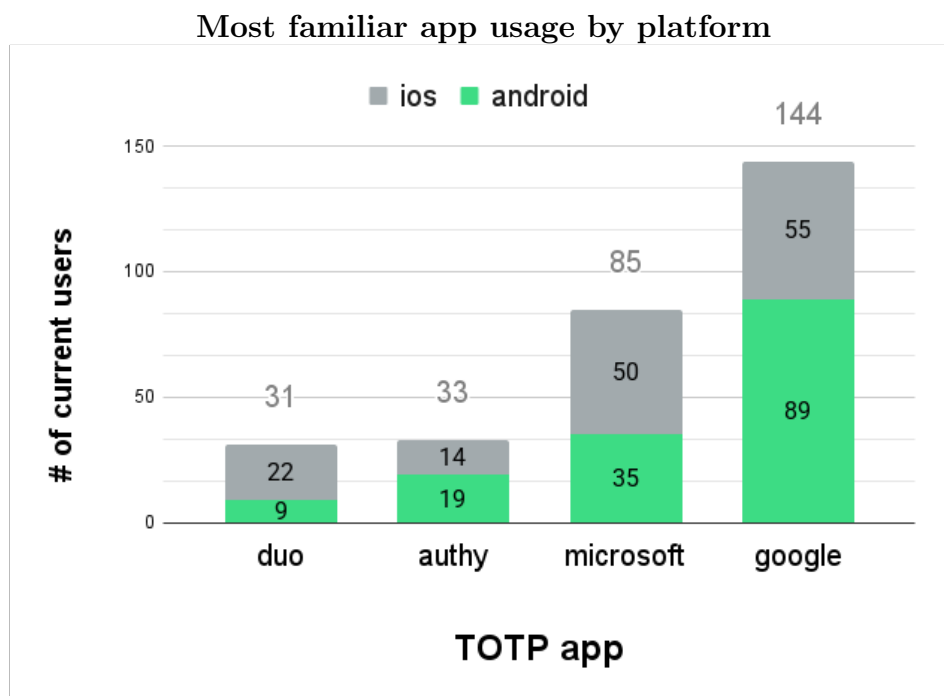


Figure 5.5: Counts of which app current users ( $n = 293$ ) were most familiar with, segmented by platform (Android & iOS).

<sup>9</sup> These participants did satisfy the criteria to be invited to the main survey, thus they likely changed their answers from the screening survey.

<sup>10</sup> Prolific rules required participants to fail two or more attention checks to be rejected on surveys longer than 5 minutes.

Demographic	Number of Participants		
Gender	Woman:	122	(37%)
	Man:	198	(60%)
	Nonbinary:	8	(2%)
	Self Describe:	0	(0%)
	Decline:	3	(1%)
Age	18-29:	87	(26%)
	30-39:	84	(25%)
	40-49:	53	(16%)
	50-59:	68	(21%)
	60-69:	32	(10%)
	70+:	6	(2%)
Education	Early:	1	(0%)
	High school:	31	(9%)
	Some college:	56	(17%)
	Associate:	37	(11%)
	Bachelor's:	135	(41%)
	Master's:	52	(16%)
	Professional:	11	(3%)
	Doctorate:	7	(2%)
Income	≤ \$20,000:	23	(7%)
	\$20k-\$39,999:	32	(10%)
	\$40k-\$74,999:	84	(25%)
	\$75k-\$99,999:	60	(18%)
	\$100k-\$149,999:	63	(19%)
	\$150k or more:	61	(18%)
	Decline:	7	(2%)
Technology Background	Yes:	159	(48%)
	No:	171	(52%)
Self-describe as tech-savvy	Disagree:	18	(5%)
	Neither:	40	(12%)
	Agree:	200	(61%)
	Strongly agree:	72	(22%)

Table 5.1: Demographics of main survey participants ( $n = 330$ ).

## 5.2.2 RQ1 - How do users perceive the risks of account lockout?

This section explores participants' basic knowledge of lockout risks while using TOTP, confidence that they could access their accounts if they lost all of their devices on which they had the TOTP app installed, and experiences with actual instances of account lockout.

### 5.2.2.1 Basic knowledge of lockout risk

In Section 5.1.2.1, we described the vignette used to quiz all users about their knowledge of account lockout risks. The majority of all users (207, 63%) *correctly answered no*, Alice would not be able to generate TOTP codes for her account after losing her device. A small group of users (55, 17%) said they did not know the answer. About one quarter of all users (68, 21%) *incorrectly answered yes*, that Alice would be able to generate the TOTPs required to login to her account.

### 5.2.2.2 Account recovery confidence

After being prompted to open the TOTP app on their device, current users ( $n = 293$ ) were asked how important they considered the accounts that were registered in the TOTP app. The overwhelming majority of current users (264, 90%) said that their registered accounts were important, with over half stressing that they were *very* important; important (114, 39%) and very important (150, 51%).

We assumed that people would want to maintain access to these important accounts, however only about half of current users (167, 57%) believed that they likely *would* be able to regain access to all of their accounts if they lost all of the devices on which the TOTP app was installed. The survey asked users to explain in their own words how they would attempt to regain access to those accounts. To avoid priming participants, this open-ended question was asked early in the survey before any other question mentioned backups. The Kupper-Hafner concordance statistic for this question was 0.899, indicating excellent inter-rater reliability. The top 10 labels applied during qualitative analysis are listed in Table 5.2.

The most common recovery approach described by current users was to reinstall the TOTP app on a different device (e.g., new device, old personal device, friend's device, etc) to access their TOTP codes (52, 18%). Of those 52 who said they would reinstall, 37 (71%) had cloud backups enabled and 13 (25%) had them disabled. Reinstalling the app could help those with cloud backups restore their ability to generate TOTP codes, but would not help those with cloud backups disabled. Several participants mentioned reinstalling the app as their only approach, such as P13 who had cloud-backups disabled ("*Re-download the app*") and P81 who had them enabled ("*By recovering my Authy data using my phone number login with my backup key on a new device.*"). Another group directly stated that they would utilize the cloud-backup option in their TOTP app (29, 10%) without mentioning the need to reinstall the app. Good examples of this approach were P193 who said "*Authenticator has*



Recovery strategies	#	(%)	Explanation
reinstall_totp_app	52	18%	They think that they can access their TOTP codes by installing the TOTP app on a different device (e.g., new device, old personal device, friend's device, etc).
reset_password	49	17%	They think they could login after resetting their password via the Forgot Password link or similar.
recovery_codes	41	14%	They would use an account recovery code to login to their account.
help_from_website_last_resort	38	13%	They said that they would reach out to customer support at the website as a last resort.
alt_auth_somewhat	36	12%	They think they can login on their own without TOTP, but do not explain how they would do this. For example "some other way", "another option", etc.
login_with_password_only	34	12%	They think that they can login to their accounts with just a password. It is likely unclear how they would login with just a password when they have TOTP enabled.
help_from_website_first_resort	33	11%	They said that they would reach out to customer support at the website as their first or only strategy.
totp_cloud_backups	29	10%	They would use cloud backups from their TOTP app, or they have cloud backups enabled and said they would login to their cloud account (e.g., Google).
alt_auth_sms	28	10%	They think they can login by having the website send them a text message.
alt_auth_email	27	9%	They think they could login by having the website send them an email.

Table 5.2: Top 10 recovery approaches that current users ( $n = 293$ ) said they would try. Descriptions could have multiple labels. Kupper-Hafner IRR = 0.899.

*a back up feature that I am using*" and P81 who said *"By recovering my Authy data using my phone number login with my backup key on a new device."*

Many current users mentioned recovery strategies that involved their passwords. Some clearly misunderstood the risk of lockout when using TOTP and thought that they could login to their accounts with just their password (34, 12%). The second most common

recovery approach that current users described was similar: reset their password (49, 17%) by using the “Forgot Password” link or similar. Even if they were successful in resetting the account password, it is not clear how these participants would access their accounts since TOTP would, presumably, still be enabled. However, as mentioned previously, researchers have documented the ability to login to accounts without the primary authenticator [6, 43].

Many people said they would leverage standard recovery and authentication methods, such as account recovery codes (41, 14%), SMS 2FA (28, 10%) and getting a login code via email (27, 9%). Some users mentioned they would login “somehow” (36, 12%), but did not explain what they would actually do, like P233 who said “*I would access online and click the option to login another way.*”

Participants often described a few recovery approaches before saying they would reach out to customer support if nothing else worked. In contrast, another group of current users (33, 11%) said that they would reach out to customer support as their first—and almost always *only*—recovery approach. P210 exemplified this by saying “*I’m actually not sure. I would probably contact customer support for each website/account I am trying to login.*” Interestingly, of those 33 participants, more than half (20, 61%) actually had cloud-based backups *enabled*. If they could access their cloud backups, then they would likely be able to login to their accounts without contacting customer service.

### 5.2.2.3 Experiences with lockout

Of all participants, there were 202 (61%) who said they had not been able to provide a TOTP code within a short window of being prompted (see Section 5.1.2.1). Of those potential lockouts, 98 (49%) were disregarded because their only reported reason for not being able to provide the code when prompted was that their device was not *immediately* available, but they could have gotten a TOTP code if they wanted to. For example, the phone with their TOTP app was in the other room and they chose not to get off the couch, they left it in the car, it had a dead battery, etc. The remaining 104 (51%) participants reported at least one additional reason and were considered to have experienced account lockout in the past (“Lockouts”). The 104 Lockouts composed just under one third (32%) of all users.

Two-tailed Fisher’s exact tests revealed statistically significant differences in the rate of account lockout among several groups. Table 5.3 shows the contingency tables and corresponding p-values.

Current users had encountered lockout more than 2 times less than previous users ( $p = 0.0011$ ). We suspected that previous users may have disabled TOTP on their accounts, switched to a different TOTP app, or made other changes specifically because of their experience with account lockouts. However, when the 21 previous users who had experienced lockout (see Table 5.3) were asked why they stopped using the TOTP app(s) they had used in the past, very few mentioned account lockout as a reason for the change.

Participants who correctly answered the vignette in Section 5.2.2.1 were considered to be aware of the risk of account lockout when using TOTP, while users who answered that question incorrectly were considered unaware. Risk aware users had experienced account

	Lockout? Yes	Lockout? No
Current Users	83 (28%)	210 (72%)
Previous Users	21 (57%)	16 (43%)

(a) Current and previous users ( $p = 0.0011$ ).

	Lockout? Yes	Lockout? No
Risk aware	30 (44%)	38 (56%)
Risk unaware	74 (28%)	188 (72%)

(b) Participants aware and unaware (see Section 5.2.2.1)  
of lockout risks when using TOTP ( $p = 0.0185$ ).

Table 5.3: Two-tailed Fisher’s exact tests showed significant differences in lockout rates among various groups ( $N = 330$ ). The p-values are listed below contingency tables (a) and (b).

lockout over 50% more than risk unaware users ( $p = 0.0185$ , see Table 5.3). It could be the case that risk aware participants were able to answer the question correctly *because* they had personally experienced lockout in the past and had been unable to recover their TOTP codes.

The most common reason among Lockouts that they could not provide the TOTP code was that they had purchased a new phone (43, 41%). Presumably, the TOTP app was installed on their old phone and they did not think to transfer the TOTP data to the new phone before losing access to the old phone. Lockouts also encountered other issues with their phone/mobile device, including: broken (20, 19%), lost (14, 13%), and stolen (5, 5%). Some lockouts (29, 28%) had also uninstalled the TOTP app from their device. (Note that participants selected all lockout reasons that applied to their situation, so the total can be over 100%).

The vast majority of Lockouts were able to regain access to at least some of their accounts quickly; 52 (50%) within one hour and 47 (45%) within one day. A smaller group was locked out of one or more accounts for one week (14, 13%) or more (3, 3%). Unfortunately, 6 (6%) Lockouts reported that they were never able to regain access to one or more of their accounts.<sup>11</sup>

Figure 5.6 shows the account recovery strategies that Lockouts used when trying to regain access to their account(s). We wanted to minimize the need for memory recall in this question, so we provided a predefined list of the most prevalent recovery methods and an option to write-in an answer. The basic descriptive statistics for the number of recovery strategies that Lockouts tried were: min (1), max (6), median (2.0), mean (2.07).

When asked whether their experience with account lockout caused by not being able to provide a TOTP code resulted in any changes to their account login settings, about one-third

<sup>11</sup>Note that participants could select more than one lockout duration, so the total can be over 100%.

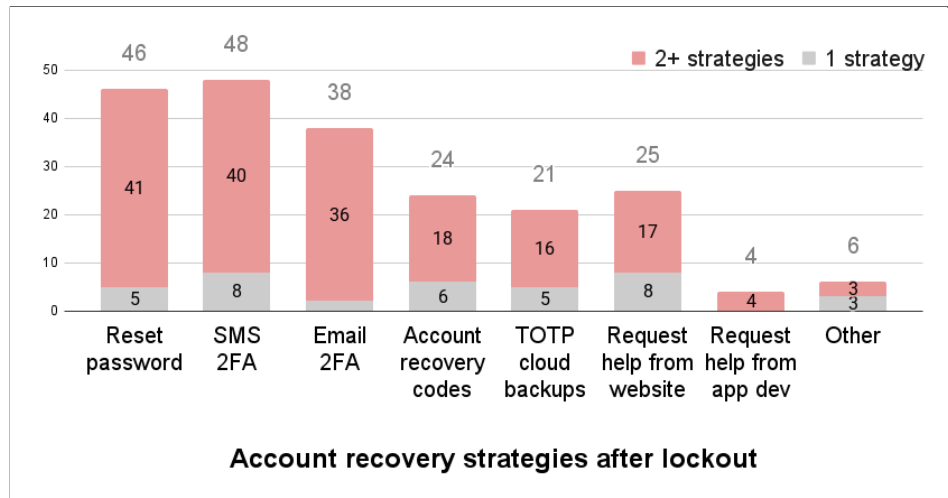


Figure 5.6: The account recovery strategies that Lockouts ( $n = 104$ ) used when trying to regain access to their account(s), grouped by people who used a single strategy and people who used multiple strategies.

of Lockouts (38, 37%) said they made no change. A little less than half of Lockouts (41, 39%) said that they reset their account password, but it is not clear why they took that action. Interestingly, password reset was also a common recovery strategy that current users said they *would* try (see Section 5.2.2.2) if they lost their devices **and** the most common recovery strategy that Lockouts *did* try in practice (see Figure 5.6). Removing TOTP from an account when the user cannot provide the TOTP code defeats the security of TOTP, but some sites have displayed this insecure behavior in practice [6, 43]. Perhaps, some websites also required users to reset their password after using certain recovery mechanisms. Some users (17, 16%) reported temporarily disabling TOTP on their account(s), which could indicate that they disabled and re-enabled TOTP to get a new setup QR-code to scan in a new app. Experiencing account lockout did cause a small group (9, 9%) to permanently disable TOTP on one or more of their accounts.

### 5.2.3 RQ2 - What precautions do users take to reduce the risk of account lockout?

In various open-ended questions throughout the survey, participants mentioned that they would use (or had used) various other methods of 2FA when trying to login to their accounts if they ever lost access to their TOTP app (see Sections 5.2.2.2 and 5.2.2.3). Receiving a login code via email or SMS were popular strategies, but ones that rely completely on the authentication and backup features supported on a site-by-site basis.

This section focuses on the five backup strategies discussed in Section 5.1.2.2 that directly apply when protecting personal, non-work accounts with TOTP. Of the 5 backup strategies

that we asked current users about, the basic descriptive statistics for how many each current user actually utilized were: min (0), max (5), median (2.0), mean (1.97) strategies.

The results in this section apply only to current users (293, 89%), unless specifically stated otherwise.

### 5.2.3.1 Cloud-based backups

Even though all of the TOTP apps included in the survey supported cloud-based backups, most users (166, 57%) did not know whether this feature was supported in their app or not. A large portion knew that it was supported (114, 39%) and only a small group incorrectly thought that it was not supported (13, 4%).

Current users were informed that cloud-based backups were supported in their app and asked whether they believed the feature was enabled or disabled. A plurality of current users (142, 48%) believed cloud backups were likely enabled, while almost a third were unsure (89, 30%) and the rest (62, 21%) believed it was disabled.

Microsoft Authenticator and Authy had the cloud backups feature disabled by default. On first install, Google Authenticator displayed a welcome prompt with a large blue button that encouraged users to login to their Google account and a small link “Use Google Authenticator without an account,” but neither option was explained; users were not informed that logging into their Google account would automatically enable cloud backups. Duo Mobile made a significant change to their cloud backups feature in summer 2024 and switched from opt-in to opt-out.

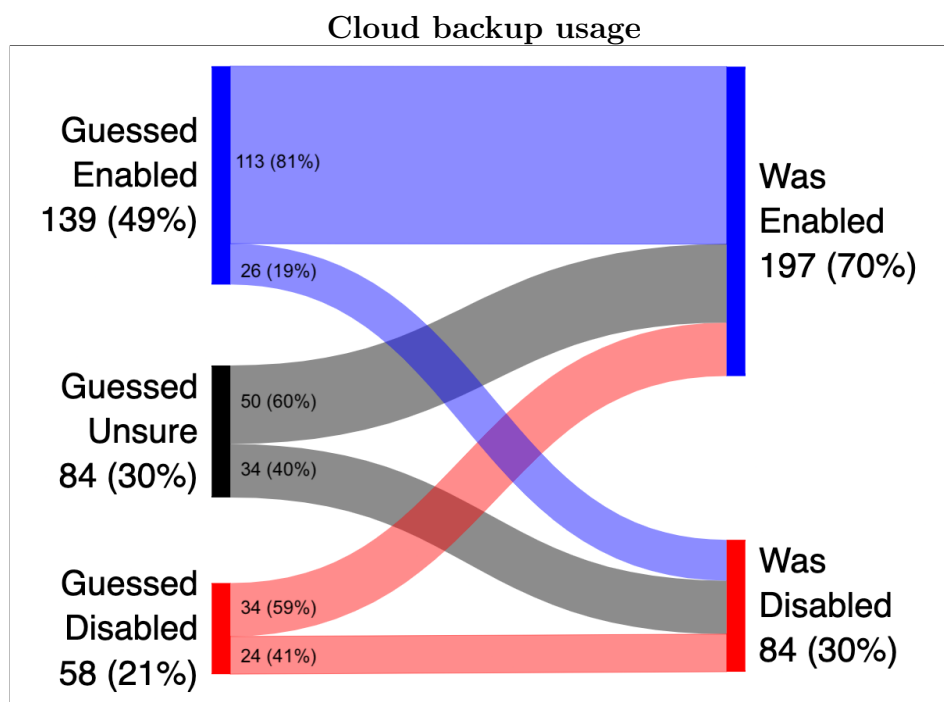


Figure 5.7: Cloud backups guess and actual usage among current users ( $n = 293$ ).

After guessing, current users followed a detailed set of instructions (see Section 5.1.2.2) to determine whether cloud-based backups were, in fact, enabled or disabled in their app. Unlike their guesses, most (197, 67%) had cloud-based backups enabled in their app, while 84 (29%) had it disabled. Figure 5.7 shows how accurate the guesses of current users were. Most of the participants who guessed *unsure* or *disabled* actually had cloud backups turned on. Only a small group of those who guessed *enabled* actually had the feature turned off. Figure 5.8 shows the usage of cloud backups segmented by TOTP app.

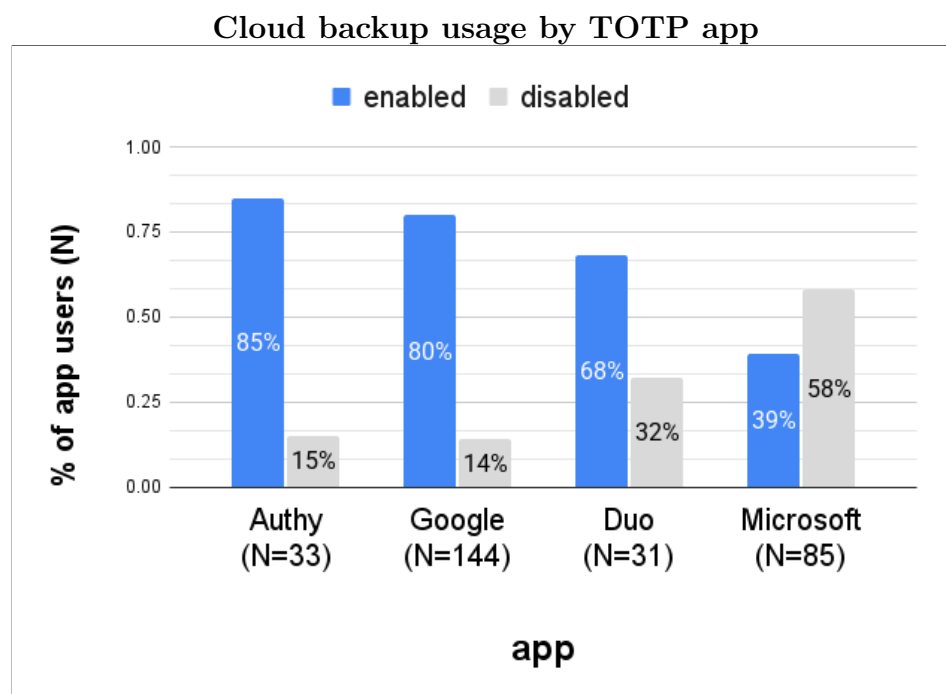


Figure 5.8: Cloud backups usage among current users ( $n = 293$ ), segmented by TOTP app. Note: some users did not report enabled/disabled.

**Cloud-based backups were *enabled*** This subsection focuses on the majority of current users (197, 67%) who had cloud-based backups enabled in their app (“enabled cohort”).

Among the enabled cohort, a small group (21, 11%) did have experience recovering their TOTP codes using cloud-based backups, but most (164, 83%) had never used the feature. Almost every single participant in the enabled cohort (184, 93%) reported that they would likely leave the cloud-based backup feature enabled in their app moving forward.

The enabled cohort was asked to explain in their own words why they chose to enable cloud-based backups. The Kupper-Hafner IRR for this question was 0.926, indicating excellent inter-rater reliability. Table 5.4 shows the top 10 labels applied during qualitative analysis.

Many cohort members (61, 31%) said that they enabled cloud backups as a general precaution, but did not explain further. For example:

- “*To have a backup in case of emergency,*” P202
- “*so I always have access,*” P244
- “*Seemed like a good idea just in case,*” P18
- “*I don’t want to lose my information. I’ve had it for a lot of years.,*” P95

However, it was most common for members of the cohort to enable cloud backups as a precaution against issues with their devices (70, 36%). Most issues were categorized into concern that their phone could become lost, stolen, broken, or that they could get a new device. For example:

- “*Hopefully I can get back into my accounts if I lose my phone,*” P149
- “*Just in case my device is lost or broken, I have a way to get my 2FA accounts set back up in the event that there are no other ways to recover them. Also, having the cloud backup is handy when swapping devices.*” P51
- “*In case my phone is lost, broken, stolen, or I need to change devices. I have very personal accounts (including my personal email, bank, mortgage company, etc.) secured using Google Auth and it would be catastrophic if I lost access.*” P225

Almost one-third of the cohort (55, 28%) described how they were surprised to learn that cloud backups were, indeed, enabled in their app and a small group said that they believed the cloud backups feature was enabled by default (22, 11%). P10 used Google Authenticator and said “*I don’t remember enabling it. I assume it was enabled by default.*” P128 used Duo Mobile and said “*It was automatically on. I didn’t choose it.*”

**Cloud-based backups were *disabled*** This subsection focuses on the minority of current users (84, 29%) who had cloud-based backups disabled in their app (“disabled cohort”).

The majority of the cohort (58, 69%) said that they had never previously enabled cloud-based backups, but some were unsure (17, 20%). Over half of the cohort (49, 58%) said that they likely *would* enable the cloud-based backup feature in the future.

The disabled cohort was asked to explain in their own words why they chose to disable (or never enable) cloud-based backups. The Kupper-Hafner IRR for this question was 0.938, indicating excellent inter-rater reliability. Table 5.5 shows the top 10 labels applied during qualitative analysis.

The most popular reason (by a wide margin) among the disabled cohort for not enabling cloud backups was that they simply did not know that the feature existed (33, 39%). In fact, some users said that they had already enabled cloud backups (9, 11%) once the survey informed them that the feature existed, while others said they might do the same (9, 11%) after learning more about the feature. Other cohort members said that they knew about the feature, but thought that it was already enabled (7, 8%).

A group of users asserted that the feature must have been disabled by default (13, 15%). The majority of them were most familiar with Microsoft Authenticator and Authy, which did have cloud backups disabled by default, but a few mistakenly thought that Google and Duo Mobile were off by default (see Section 5.2.3.1).

Various concerns caused some cohort members to disable (or never enable) cloud backups. Some had generic concerns about “the cloud” (6, 7%), such as P139 who said “*i don’t really trust the cloud*” and P62 who said “*I prefer to keep everything stored locally.*” P131



Reason enabled	#	(%)	Explanation
precaution_general	61	(31%)	They enabled the feature as a precaution. It may be undefined what they were worried about. For example, “just in case”, “you never know”, “it seemed like a good idea”, etc.
surprised	55	(28%)	They were surprised to learn that the feature was enabled.
precaution_device_lost	46	(23%)	They mentioned that they could lose their device, or generally lose access to it.
on_by_default	22	(11%)	They said that cloud backups were (likely) enabled by default.
precaution_device_broken	19	(10%)	They mentioned that their device could break.
precaution_device_new	15	(8%)	They mentioned getting a new device.
multiple_device_use	10	(5%)	They mentioned something about using multiple devices. For example, keeping TOTP codes in sync across devices, accessing things on multiple devices, etc.
precaution_device_general	10	(5%)	They mentioned that “something” could happen to their device(s).
convenience	10	(5%)	Participant believes cloud-based backups are convenient for their situation.
account_importance	9	(5%)	They specifically mention the importance of the accounts that they have registered in their TOTP app.

Table 5.4: Top 10 reasons the enabled cohort ( $n = 197$ ) enabled cloud backups. Each response could have multiple labels. Kupper-Hafner IRR = 0.926.

exemplified those who had security concerns (5, 6%) and said “*I disabled it mainly for security or if some how my backup got hacked.*” We were surprised to see several cohort members explain that they had concerns about limits on storage space in their cloud accounts (7, 8%); either they were already out of storage space, (“*I didn’t realize I didn’t have it activated because of lack of icloud space*”), or they were worried about how much space backups would take (“*I didn’t want it to store a lot of things and run out of storage,*” P117).

A group of users asserted that they did not need cloud backups (9, 11%); some clearly did not understand the risk of losing their TOTP codes (“*I don’t understand why it’s necessary since I don’t think I need backups,*” P249) while other said they did not really care about the accounts registered in their app. A smaller group of users did care about backups generally, but said that they were confident in other backup precautions that they had already taken (5, 6%). P153 explained their recovery approach in some detail: “*I have cloud backups disabled*

Reason disabled	#	(%)	Explanation
did_not_know_it_existed	33	(39%)	They did not know that the cloud backups feature existed.
off_by_default	13	(15%)	They believe the cloud backups were disabled by default and they just never changed it.
not_needed	9	(11%)	They said that they do not need it. The reason that they believe they do not need it might be unexplained.
enable_might	9	(11%)	They mention that they might enable it (e.g., consider it, look into it, maybe turn it on, etc).
enable_did	9	(11%)	They said that they enabled cloud backups while taking the survey.
concern_cloud_storage_space	7	(8%)	They mention that they are concerned about storage space in their cloud account. For example, not having enough storage space, or running out of storage space if they enable cloud backups.
thought_enabled	7	(8%)	They were previously aware of the feature and thought that it was already enabled.
concern_cloud_generic	6	(7%)	They are against the cloud for some reason. For example, they do not like it, do not trust it, do not use it, etc.
confident_in_other_backups	5	(6%)	They are confident that the other backup strategies they setup would work to avoid lock-out.
concern_security	5	(6%)	They believe cloud-based backup option may present security vulnerabilities.

Table 5.5: Top 10 reasons the disabled cohort ( $n = 84$ ) disabled (or never enabled) cloud backups. Each response could have multiple labels. Kupper-Hafner IRR = 0.938.

*to lower the chance of a security breach. I have google authenticator on multiple devices and it is highly unlikely that I lose all of them all at the same time. Even if I did, I can still access the backup codes which are securely stored.”*

### 5.2.3.2 QR backups

Support for QR-based backups is uncommon among the most popular TOTP apps (see Chapter 4) and, within our dataset, Google Authenticator was the only app that supported this feature. About half of the current users (144, 44%) were most familiar with the Google Authenticator app (“Google users”).

Only half of the Google users (76, 53%) knew that the QR-based backups feature existed. Of those who knew it existed, only 48 (63%) had actually used the feature. Of those who had used it, only 31 (65%) agreed that all of their accounts were backed up to another device. It seems reasonable to conclude that relying solely on manual backups via QR-code is not a viable solution for most people in practice, since those 31 users represent just 22% of all Google users. Accounts that are not backed up manually could face a higher risk of lockout, unless other backup options are also available.

### 5.2.3.3 Multiple devices

The majority of current users said that they had only installed the app on a single device (152, 52%). A surprising number had the app installed on 2 devices (102, 35%), while more than 2 devices was much less common. Of those who only had the app set up on a single device, a little more than half (89, 59%) also had cloud-based backups enabled. These users may be able to recover their TOTP codes from those remote backups, even if they lost their single device.

### 5.2.3.4 Account Recovery Codes

A large portion of current users (108, 33%) reported that they did not have account recovery codes for any of their accounts in the TOTP app.

About half of the participants had saved account recovery codes while enabling TOTP; 64 (19%) were confident that they had recovery codes for *all* of their accounts, while 91 (28%) had recovery codes for at least *some* accounts.

Of the 155 (53%) participants who did save recovery codes for at least one account, the overwhelming majority (136, 88%) felt that they likely *would* be able to access those recovery codes if they ever lost access to all of the devices on which their app was installed.

The number of different locations that people chose to store recovery codes was more well-distributed than storage locations for setup QR-codes (see Figure 5.9). No storage location had a majority, but more than half of the current users stored the recovery codes in more than one place. The number of places that people stored recovery codes: min (1), max (5), median (2.0), mean (1.75).

### 5.2.3.5 Setup QR Code

We expected this strategy to be relatively rare among general TOTP users and, indeed, the vast majority of all participants (237, 81%) reported that they did not save setup QR-codes for any of their accounts. However, a small population (36, 12%) reported having saved at least one setup QR-code (“screenshot savers”); 17 (6%) saved a setup QR-code for *some* of their accounts, while 19 (6%) claimed that they had saved the setup QR-code for *all* of their accounts protected by TOTP.

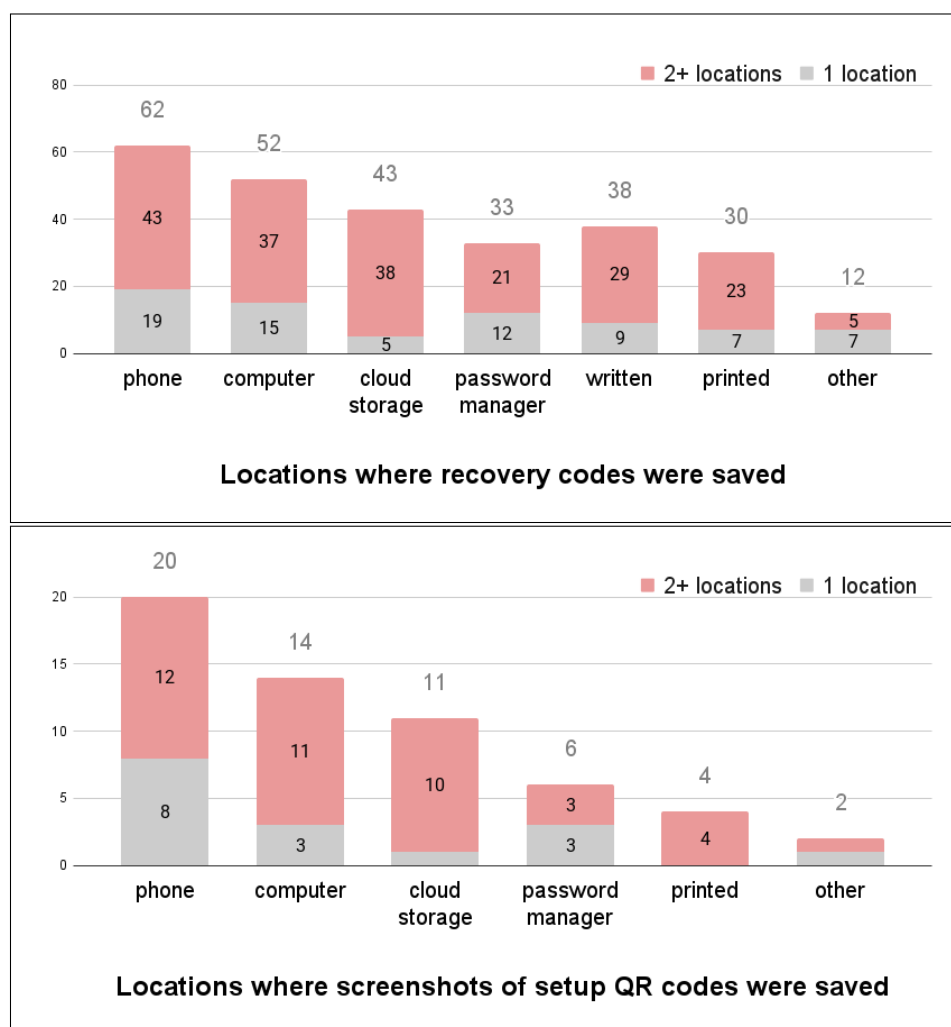


Figure 5.9: Where people stored recovery codes ( $n = 155$ ) and setup QR-codes ( $n = 36$ ), grouped by people who stored in only that location and people who also stored in additional locations. Participants could select multiple storage locations.

The screenshot savers were asked to select where they stored the setup-QR code from a list of predefined options, which can be seen in Figure 5.9. The number of places that people stored screenshots: min (0), max (3), median (1.5), mean (1.58).

The overwhelming majority of screenshot savers (28, 78%) felt that they likely *would* be able to access the saved screenshot if they ever lost access to all of the devices on which their app was installed.

Most screenshot savers (20, 56%) saved it on their phone (see Figure 5.9). While it might seem counterintuitive to store the screenshot required for recovery on the same device that would be lost in a recovery scenario, most devices automatically backup their photo libraries

to the user’s cloud storage account. If this feature is enabled, then people who lose their phone may be able to access the screenshot of the setup QR-code via their cloud account on another device.

The second most common place to store the screenshots was on the user’s computer (14, 39%). Likely, these users enabled TOTP on their account using the browser on their computer and took a screenshot during the setup process. These users gained redundancy by leveraging multiple devices. If the phone is lost, then they can install the TOTP app on a new phone and scan the screenshot on the computer. If the computer is lost, then they can still generate TOTP codes using the app on the phone.

### 5.2.3.6 High-risk users with no TOTP backups

There was a group of current users (37, 13%) who did not have any TOTP backup strategies (“high-risk users”). They had cloud-based backups disabled, had never used QR-codes to manually backup to a different device<sup>12</sup>, had the app installed on a single device, and had not saved any account recovery codes nor screenshots of setup QR-codes. If any of the high-risk users lost their single device on which the TOTP app was installed, they would unlikely be able to recover the ability to generate valid TOTP codes for their accounts. Thus, they were at the highest risk of account lockout compared to other current users who utilized one or more TOTP backup strategies.

There was no notable difference in the age distribution between high-risk users and the rest of the participant population. However, the self-reported tech-savviness among high-risk users ( $\bar{n} = 3.68$ ) was significantly less than participants who had one or more backups ( $\bar{n} = 4.06$ ) (Welch’s two-tailed t-test,  $p = 0.0024$ ). Also, there was a drastically larger percentage of high-risk users (70%) who were most familiar with Microsoft Authenticator than among the rest of the current users (23%).

**How would they attempt to recover?** About half of the high-risk users (19, 51%) claimed that they likely *would* be able to login to their accounts registered in the TOTP app if they lost all of the devices on which it was installed. Most high-risk users were likely overly optimistic given their responses when asked how they would attempt to login to those accounts (see Table 5.6). As mentioned in Section 5.2.2.2, the Kupper-Hafner IRR for this question was 0.899, indicating excellent inter-rater reliability.

The most common approach described was to ask the website they were logging into for help, many as a first (and almost always only) resort (7, 19%). For example, customer support was the only approach mentioned by P249: “*I would call customer support and explain the situation and ask for assistance in logging in without the apps.*”

The second most common approach mentioned was to reinstall the TOTP app on a new device (7, 19%). Several users said this was their only strategy: “*I would need to re-install the app*” (P24), “*Download the app on a new device*” (P181), and “*I would install the app*

---

<sup>12</sup>Only applies to Google Authenticator.

Recovery strategies	#	(%)	Explanation
help_from_website_first_resort	7	(19%)	They said that they would reach out to customer support at the website as their first or only strategy.
reinstall_totp_app	7	(19%)	They think that they can access their TOTP codes by installing the TOTP app on a different device (e.g., new device, old personal device, friend's device, etc).
login_with_password_only	6	(16%)	They think that they can login to their accounts with just a password. It is likely unclear how they would login with just a password when they have TOTP enabled.
alt_auth_somewhat	6	(16%)	They think they can login on their own without TOTP, but do not explain how they would do this. For example “some other way”, “another option”, etc.
alt_auth_email	5	(14%)	They think they could login by having the website send them an email.
help_from_website_last_resort	4	(11%)	They said that they would reach out to customer support at the website as a last resort.
unsure	4	(11%)	They said that they are “unsure” about how they would login or are generally not confident in their ability to recover.
reset_password	3	(8%)	They think they could login after resetting their password via the Forgot Password link or similar.
alt_auth_sms	2	(5%)	They think they can login by having the website send them a text message.
help_from_app_developer	2	(5%)	They would try to get help from the developer of the TOTP app.

Table 5.6: Top 10 recovery approaches that high-risk users ( $n = 37$ ) said they would try. Descriptions could have multiple labels. Kupper-Hafner IRR = 0.899.

on a new device” (P56). Reinstalling the TOTP app would not recover their TOTP codes since cloud-based backups were disabled.

Some participants said they would try to login “some other way,” but did not explain further (6, 16%). Others were more specific and said they would login with just their account password (6, 16%), but did not explain how this would be possible since TOTP was enabled on the account(s). Just a few high-risk users mentioned approaches that were more likely

to succeed (see Table 5.6), including getting a login code via email or SMS, and using a different device on which they might already be logged into certain accounts (“remember me”).

**Will they enable backups?** The vast majority of high-risk users (27, 73%) considered the accounts registered in their TOTP app to be important or very important. We assumed that they would not want to lose access to those accounts and would, therefore, choose to enable cloud-based backups since the survey previously informed them of the feature’s existence. About half of high-risk users (20, 54%) said they were likely to enable the feature. A smaller group (12, 32%) said they were still undecided, which could be easily explained; for example, they may have wanted time to research a feature that was new to them before taking action.

Curiously, there were just a few high-risk users (5, 14%) who said that they were *unlikely* to enable cloud-based backups. Two of these users (P117, P154) were concerned with the limited amount of storage space they had in their cloud account or said they did not have enough storage space. P234 said TOTP was off by default and they disliked TOTP in general: “*I didn’t actually choose it either way, it was just disabled by itself. I never bothered to look at the settings. I have the authenticator against my will.*” P249 did not understand the need for backups: “*I don’t understand why it’s necessary since I don’t think I need backups.*” And P312 did not trust cloud storage in general, saying “*Force of habit don’t like to put stuff on the cloud.*”

#### 5.2.4 RQ3 - What privacy concerns do users have about TOTP apps?

There are three key pieces of personal information for each account registered in a TOTP app that users may not want to share. The name of the website and the account username are used by the app to visually label which TOTP code goes with which account. The third field is a static secret that is used by the TOTP app to generate the 6-digit code required to login to the account.

First, we wanted to learn which data participants expected could be read from cloud-based backups created by their TOTP app. When current users ( $n = 293$ ) were asked which of the three TOTP fields they thought were collected by the *app developer* (see Figure 5.10), most thought that the app developer *could* read both the username and the website name from accounts included in their cloud-based backups. A minority expected those fields remain protected from access by the app developer. Current users were more spread on whether they believed the app developer could read the TOTP secret, with about one third each saying they did expect it could be read, did not expect it could be read, and that they did not know.

There was a significant shift in expectation when current users were asked whether they believed the same data could be read by *third-parties* (not the app developer). For each of the

three TOTP fields, only a minority of users believed that it could be read by third-parties. The TOTP secret showed the strongest belief that it *could not* be read by third-parties (175, 60%).

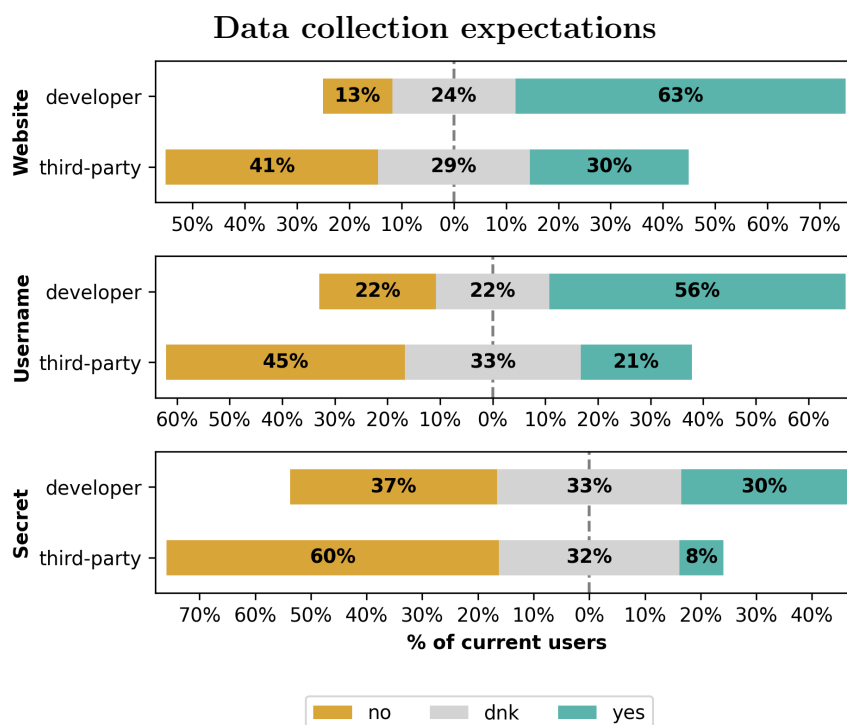


Figure 5.10: How many current users ( $n = 293$ ) believed the app developer and other third-parties could read TOTP fields from cloud backups created by their app. dnk = “I do not know”

After asking about their expectations of data collection, we wanted to learn how current users would feel if their data could *actually* be read. As we found in Chapter 4, the cloud backup mechanism in several prevalent TOTP apps allowed some or all of the three TOTP fields to be read by the app developer or other third parties. In almost every scenario, people were uncomfortable with app developers or third-parties collecting data about the accounts they had registered in their TOTP apps (see Figure 5.11). People were most comfortable with the website name being collected by the app developer. In contrast, a majority of current users were uncomfortable with each of the three TOTP fields being collected by either the app developer or third-parties. People felt most strongly about collection of the TOTP secret; more than half of current users felt *very* uncomfortable with the app developer collecting the secret (149, 51%) and almost three-quarters felt *very* uncomfortable with the secret being accessed by third-parties (208, 71%).



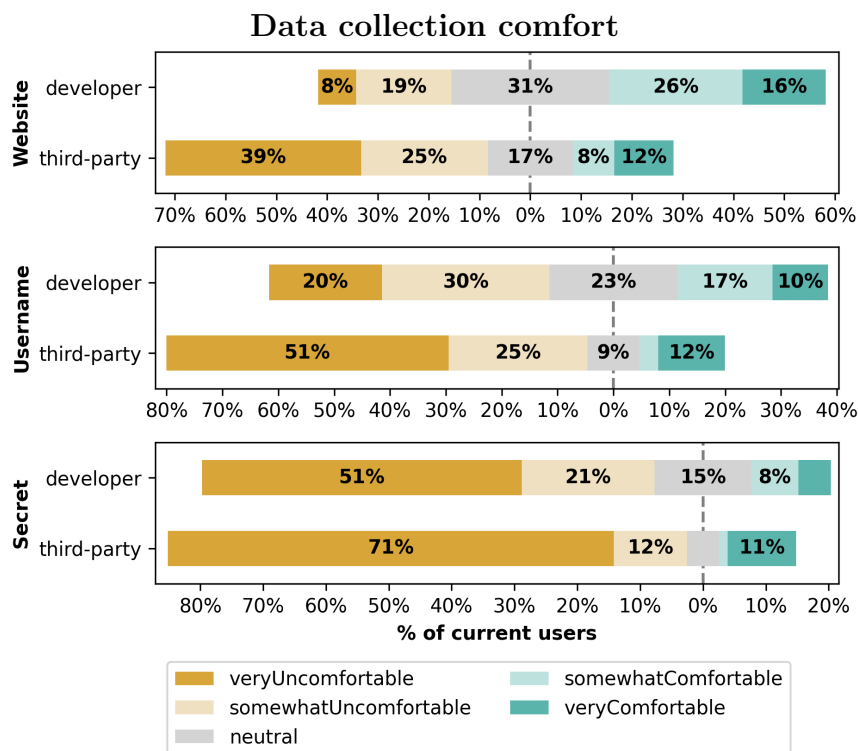


Figure 5.11: How comfortable current users ( $n = 293$ ) would be if TOTP data in cloud backups could be read by the app developer and other third-parties.

### 5.3 Limitations and future work

It is possible that participants misunderstood the text-based descriptions of the actions would take when using different methods of 2FA. However, the goal in this work was not to provide metrics on the usage rates of all methods of 2FA. Rather, it was specifically to identify users of specific TOTP apps. We are confident that we achieved this goal because the main survey explained the basic functionality of TOTP apps in a comprehension quiz, showed an example TOTP app, and verified app usage by asking users to reselect the TOTP apps they used while displaying the app name alongside the app logo and developer name. It would be exciting if future work tested the reliability of our text-based 2FA descriptions (see Appendix B.1) and also explored other modalities, such as screenshots, animations, and videos. Such work could yield metrics on which 2FA methods are most frequently used among the general population, for which there is not currently a reliable source of data.

In open ended responses, many participants mentioned that they would login to their Google account in order to recover. In retrospect, it would have been useful if the main survey asked current users which 2FA methods they had enabled on their primary email

account (e.g., Google, Microsoft, Apple, etc) so that we could analyze the risk that they would not be able to gain entry to that account during recovery.

The main survey asked participants about the accounts that they had registered in their TOTP app in aggregate. Future surveys could ask users about specific accounts registered in their TOTP app and how those accounts are actually protected. For example, do those accounts actually have SMS 2FA enabled? This would allow a deeper analysis on how users could recover accounts in practice.

# Chapter 6

## Conclusion

This conclusion distills both the technical findings in Chapter 4 and the survey results in Chapter 5 into a few key observations and arguments about the security and privacy impacts of using TOTP.

### **6.1 In practice, TOTP does not escape the vulnerabilities of SMS 2FA**

As prior work outlined in Chapter 3 shows, SMS 2FA has a long list of vulnerabilities. As a result, TOTP is often promoted as a more secure alternative. Ironically, I have been on the receiving end of this message from some websites on which I have accounts. PayPal, for example, nudged its users by email to switch from SMS 2FA to either TOTP or security keys, stating that TOTP offers “much stronger security benefits” than SMS (see Figure 6.1).

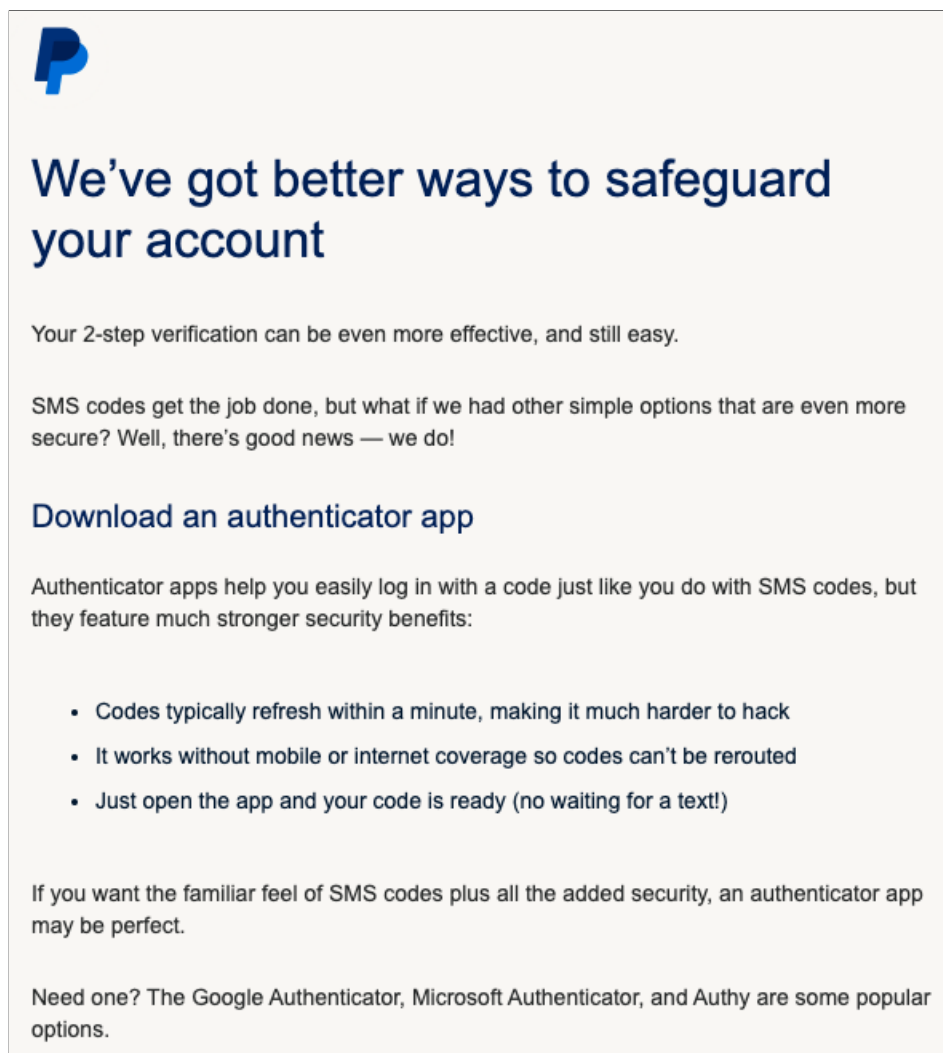


Figure 6.1: PayPal email promoting TOTP 2FA as more secure than SMS 2FA.

One claim to motivate the change was that the OTPs generated by TOTP apps are harder to hack because they rotate frequently. Put aside the fact that OTPs sent via SMS could be set to expire relatively frequently as well. PayPal also highlights—rightfully so—that OTPs are generated locally and cannot fall victim to SIM swapping and other rerouting attacks. However, any security benefits that time-based OTPs *might* provide are negated if users leave SMS 2FA enabled *in addition to* TOTP. Our findings in Chapter 5 showed that, of the 104 users who experienced account lockout while using TOTP, 48 (46%) said they tried to recover their accounts via SMS 2FA. This is only possible if the account has both SMS 2FA and TOTP enabled simultaneously. Attackers will always target the path of least resistance, so they will simply target SMS like they would have for an account protected only by SMS 2FA.

Suppose that TOTP users wanted to actually realize the security benefits of TOTP and avoid the vulnerabilities of SMS entirely. That would require them to disable SMS 2FA on their account. In this case, TOTP users are directly faced with the usability challenge of maintaining access to their TOTP secret. Results from Chapter 5 showed that 71% of current users had cloud-based backups *enabled*. While this feature may solve their usability problem of maintaining access to the TOTP secret, it brings the user back to square one. Chapter 4 showed that, in order to access cloud backups created by their TOTP app, users need to authenticate to the cloud provider using SMS, email, and passwords.

There are, of course, some users who are able to realize the promised security benefits of TOTP. For example, one could disable SMS 2FA, disable cloud-based TOTP backups, and take other backup precautions, such as storing account recovery codes or saving a screenshot of the TOTP setup QR-code. However, survey results from Chapter 5 show that people who follow all of these steps are in the minority.

**The data collected in this dissertation shows that, in practice, TOTP 2FA gets used *in addition to*, not *instead of*, SMS 2FA. Therefore, I argue that promoting TOTP 2FA as more secure than SMS 2FA is misleading in most cases. It simply does not match the reality on the ground.**

Finally, it is also important to take into consideration that companies are often motivated to get users to switch from SMS 2FA to TOTP for non-security reasons. The prime one being that text messages cost money. Each time a website sends a user an OTP via SMS, they pay a fee to their SMS provider. On the other hand, TOTP is just another web request, which is comparatively inexpensive. The most well known example of this approach is likely when Twitter restricted the option to enable SMS 2FA to only paying customers in February 2023 [112]. Figure 6.2 shows the message displayed to me on my personal Twitter account on March 22, 2023.

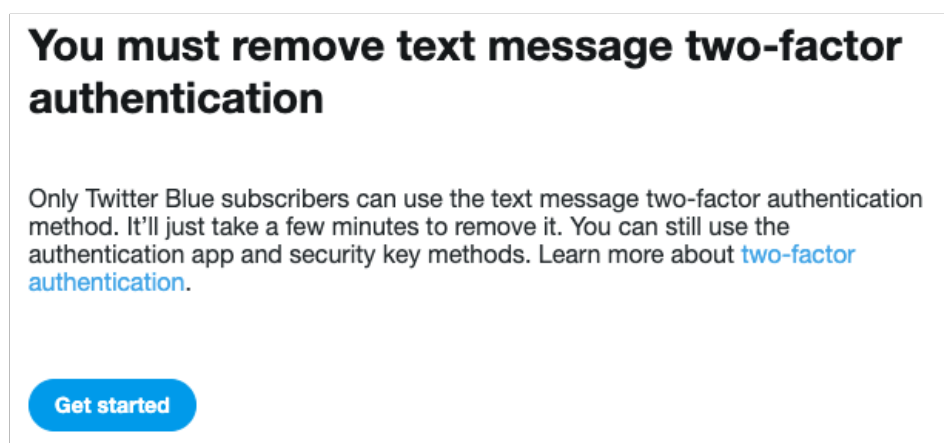


Figure 6.2: Twitter restricted SMS 2FA to paying customers only in March 2023.

## 6.2 TOTP 2FA should be retired in favor of FIDO2

As discussed throughout this dissertation, TOTP has many drawbacks. The usability challenge with backup and recovery forces most users to enable cloud backups, which reintroduces the issues with SMS 2FA, email, and passwords. TOTP is also vulnerable to phishing and social engineering attacks. Additionally, attackers could steal the TOTP secret for a user's account from both their personal devices and from the website's server. If a data breach were to occur and an attacker obtained the TOTP secrets, they would be able to generate the OTPs required to login to users' accounts.

In contrast, as discussed in Chapter 3, the FIDO2 collection of specifications is supported by all major browsers and leverages public key cryptography to provide a phishing resistant authentication mechanism. Since the server only stores a public key, attackers would not gain the ability to login to users' accounts if they breached authentication databases.

Passkeys do suffer from the same backup and recovery issues as TOTP. If the user loses their device on which the private key is stored, they will not be able to complete the login challenges for their accounts. The current industry solution is to backup these private keys on users' cloud storage accounts (e.g., Google, Apple, and Microsoft accounts). Just like TOTP, when someone loses their device and actually needs to recover, they will fall back to SMS 2FA, email, and passwords to login to their cloud accounts. Since the backup and recovery issue is present in both mechanisms, developers should favor FIDO2 over TOTP because of the other increases in security, especially phishing resistance.

The FIDO Alliance<sup>1</sup>, the primary organization promoting passkeys development and adoption, could use the empirical evidence in this dissertation to help clarify misconceptions about TOTP and strengthen the case for adopting passkeys. The motivation for developers to implement TOTP likely includes the goal of providing better security than SMS 2FA. Many developers may consider TOTP "good enough," since it is often promoted as more secure than SMS 2FA and it is easier to implement than FIDO2. While these developers may believe they are achieving the goal of improved security by moving from SMS 2FA to TOTP, in practice, they are simply introducing additional layers of SMS 2FA, email, and passwords. The FIDO Alliance can use this research to highlight that TOTP achieves a much smaller practical security improvement than most developers realize. If developers knew this, they might be more willing to skip TOTP altogether and adopt passkeys as the better long-term solution.

## 6.3 TOTP apps should be improved during the migration to FIDO2

The survey results from Chapter 5 show that most people using TOTP to protect their personal, non-work accounts did have cloud backups enabled. As a result, these users are

---

<sup>1</sup>[fidoalliance.org](https://fidoalliance.org)

likely exposed to some of the security and privacy issues uncovered by the reverse engineering work from Chapter 4. As I discussed in Section 6.2, passkeys address many of these security and privacy issues, but the technology is still relatively new. Adoption at the scale of the internet is slow and it will likely be many many years before passkeys become ubiquitous and replace existing alternatives like TOTP. For comparison, web certificates were introduced in the 1990s, yet it took more than 2 decades for the vast majority of web traffic to become encrypted [105]. The hundreds of millions of people who use TOTP apps *today* deserve to have access to apps that have the smallest negative impact to their security and privacy as possible. Therefore, the security and privacy issues in TOTP apps should be addressed even though TOTP, in the long term, should be retired.

As discussed in Chapter 5, most TOTP users expect that the app developer can read certain TOTP fields from cloud backups, but that they are uncomfortable with that data collection. TOTP apps need to improve their security mechanisms and be more transparent when data is collected so that users can make informed decisions.

TOTP apps that implement a backup mechanism with end-to-end encryption (e2ee) should encrypt all of the TOTP fields (website, username, and secret) so that the app developer does not have the technical means to access any of the backup data. If data is not end-to-end encrypted, then once it leaves the user’s device, there are no guarantees what will happen to it. Users must trust companies that store and process this data. Take, for example, Microsoft’s response to our responsible disclosure that they had the technical means to read the contents of cloud backups created by the Microsoft Authenticator Android app. Rather than explain why users of the iPhone app were offered better privacy (see Section 4.2.3.4, they confirmed the backup mechanism in their Android app was “by design” and highlighted their internal security mechanisms (see Section 4.3.3. Most companies, Microsoft included, probably do have the best of intentions. However, mistakes happen, insider threats exist, and data breaches occur. There is no need for this trust to exist when simple changes can provide cryptographic assurance that data from cloud backups cannot be read.

All TOTP apps, especially those that do not implement e2ee, such as ones that rely on the default Android Backup system, need to drastically improve their transparency about what data they have the capability to read from cloud backups. Many of the users surveyed in Chapter 5 were unaware that the cloud backup feature existed and that it was enabled. Furthermore, one third of current TOTP users were unsure whether the app developer could read data from their cloud backups or not. This implies that many TOTP users did not provide informed consent to have their data to be processed and stored remotely. TOTP apps must do a better job informing users when cloud backup features are enabled, the benefits of doing so, and the potential privacy impacts that may occur.

### 6.3.1 Implications for regulators

The privacy exposures uncovered in this research have direct implications for regulators, particularly those involved in shaping data protection laws and consumer protection frameworks. As shown in Chapter 5, many users in our study were unaware that features like

cloud backup were enabled in their TOTP apps, and as a result, that their data was being uploaded and processed remotely. This means that users were not providing informed consent for how their authentication data was being handled.

In jurisdictions like the European Union, this raises serious concerns under frameworks such as the General Data Protection Regulation (GDPR), which mandates that, when user consent is the lawful basis for data collection and processing, that consent must be freely given, specific, informed, and unambiguous.<sup>2</sup> If users are unaware that a cloud backup feature is even active—let alone what it stores and who can access it—then the consent they have supposedly given does not meet that standard.

In the United States, where there is still no comprehensive federal privacy law, consumer protection regulations and emerging state privacy laws create a patchwork to regulating privacy harms, such as the ones we identified. This patchwork makes it more challenging to hold businesses accountable and protect users in situations where their data may be collected without their knowledge or informed consent.

These findings suggest that stronger regulatory scrutiny is warranted, both to enforce existing protections in places like the EU and to catalyze the development of equivalent protections in the U.S. Privacy regulators should not only focus on data minimization and access control, but also on transparency in feature defaults and the comprehensibility of privacy-related user interfaces. Without such regulation, users may continue to face significant privacy risks without even realizing it.

### 6.3.2 The need for ongoing monitoring to support the public

The findings in this dissertation should be understood as a snapshot in time. TOTP apps are not static products—developers regularly change features, security defaults, and data handling practices. As such, conclusions about their usability and privacy implications may become outdated as these tools evolve. For the hundreds of millions of people who currently rely on TOTP apps, this means that a one-time decision to use a particular app may not be sufficient. Ongoing monitoring is essential, both by researchers and by users themselves, to evaluate whether the app they are using continues to meet their security, privacy, and usability needs.

This is especially important given that many of the most consequential behaviors in these apps—such as whether cloud backup is enabled by default, or what data is uploaded during sync—often occur in the background and without clear user awareness. An app that appeared privacy-preserving at one point may silently change its practices in a later update. Without sustained oversight and transparency, users are left to rely on assumptions that may no longer hold true.

---

<sup>2</sup>The GDPR provides 6 lawful bases for data processing, including: (1) consent; (2) contractual necessity; (3) legal obligation; (4) vital interests; (5) public interest; and (6) legitimate interests. In this context, the clear basis would be user consent. Given the relationship between the app user and the app developer, the other bases almost certainly would not apply.



To help address this gap, more public-facing dissemination of research is needed. Our work in Chapter 4 was featured in Wirecutter [30], which provided a valuable opportunity to reach a broader audience beyond academic and developer circles. Coverage like this plays a critical role in helping the general public make informed decisions about their authentication tools. More work is needed to translate technical findings into accessible guidance so that users can periodically re-evaluate whether their current 2FA app is still aligned with their values and threat models—or whether it may be time to switch to make a change.

# Bibliography

- [1] *2FA Directory*. Nov. 2021. URL: <https://2fa.directory>.
- [2] Jacob Abbott and Sameer Patil. “How Mandatory Second Factor Affects the Authentication User Experience”. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI ’20. Honolulu, HI, USA: Association for Computing Machinery, 2020.
- [3] Elham Al Qahtani, Lipsarani Sahoo, and Mohamed Shehab. “The Effectiveness of Video Messaging Campaigns to Use 2FA”. In: *International Conference on Human-Computer Interaction*. Springer. 2021, pp. 369–390.
- [4] Yusuf Albayram, Mohammad Maifi Hasan Khan, and Michael Fagan. “A study on designing video tutorials for promoting security features: A case study in the context of two-factor authentication (2fa)”. In: *International Journal of Human-Computer Interaction* 33.11 (2017), pp. 927–942.
- [5] Noura Alomar, Mansour Alsaleh, and Abdulrahman Alarifi. “Social authentication applications, attacks, defense strategies and future research directions: a systematic review”. In: *IEEE Communications Surveys & Tutorials* 19.2 (2017), pp. 1080–1111.
- [6] Sabrina Amft et al. “We’ve Disabled MFA for You: An Evaluation of the Security and Usability of Multi-Factor Authentication Recovery Deployments”. In: *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS ’23)*. 2023, pp. 3138–3152. DOI: 10.1145/3576915.3623180. URL: <https://doi.org/10.1145/3576915.3623180> (visited on 04/17/2025).
- [7] *Announcement of the Future Requirement to Enable Multi-Factor Authentication (MFA)*. Mar. 2021. URL: <https://help.salesforce.com/s/articleView?id=000356005&type=1> (visited on 06/07/2022).
- [8] Apple. *Passkeys Overview - Apple Developer*. URL: <https://developer.apple.com/passkeys/> (visited on 12/08/2022).
- [9] *Back up user data with Auto Backup*. Mar. 2022. URL: <https://web.archive.org/web/20220421053001/https://developer.android.com/guide/topics/data/autobackup> (visited on 05/25/2022).

- [10] Giuseppe Bianchi and Lorenzo Valeriani. “Time Is on My Side: Forward-Replay Attacks to TOTP Authentication”. In: *Security and Privacy in Social Networks and Big Data*. Vol. 14252. Lecture Notes in Computer Science. Springer, Aug. 2023, pp. 109–126. DOI: 10.1007/978-981-99-5177-2\_7. URL: [https://doi.org/10.1007/978-981-99-5177-2\\_7](https://doi.org/10.1007/978-981-99-5177-2_7) (visited on 04/18/2025).
- [11] A. Biryukov, D. Dinu, and D. Khovratovich. “Argon2: new generation of memory-hard functions for password hashing and other applications”. In: *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE. 2016, pp. 292–302.
- [12] Joseph Bonneau et al. “Secrets, lies, and account recovery: Lessons from the use of personal knowledge questions at google”. In: *Proceedings of the 24th international conference on world wide web*. 2015, pp. 141–150.
- [13] Joseph Bonneau et al. “The quest to replace passwords: A framework for comparative evaluation of web authentication schemes”. In: *IEEE Symposium on Security and Privacy*. 2012, pp. 553–567.
- [14] John Brainard et al. “Fourth-Factor Authentication: Somebody You Know”. In: *Proceedings of the 13th ACM Conference on Computer and Communications Security*. CCS ’06. Alexandria, Virginia, USA: Association for Computing Machinery, 2006, pp. 168–178. ISBN: 1595935185. DOI: 10.1145/1180405.1180427. URL: <https://doi.org/10.1145/1180405.1180427>.
- [15] Dave Childers. *State of the Auth 2021*. Tech. rep. Archived version. Duo Security, Sept. 2021. URL: <https://web.archive.org/web/20250428192211/https://duo.com/assets/ebooks/state-of-the-auth-2021.pdf> (visited on 04/28/2025).
- [16] Stéphane Ciolino, Simon Parkin, and Paul Dunphy. “Of Two Minds about Two-Factor: Understanding Everyday {FIDO} U2F Usability through Device Comparison and Experience Sampling”. In: *Fifteenth Symposium on Usable Privacy and Security ({SOUPS} 2019)*. 2019, pp. 339–356.
- [17] Jessica Colnago et al. “It’s not actually that horrible: Exploring Adoption of Two-Factor Authentication at a University”. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM. 2018, p. 456.
- [18] Lorrie Cranor. *Your mobile phone account could be hijacked by an identity thief*. July 2016. URL: <https://www.ftc.gov/news-events/blogs/techftc/2016/06/your-mobile-phone-account-could-be-hijacked-identity-thief>.
- [19] Sanchari Das, Andrew Dingman, and L Jean Camp. “Why Johnny doesn’t use two factor a two-phase usability study of the FIDO U2F security key”. In: *International Conference on Financial Cryptography and Data Security*. Springer. 2018, pp. 160–179.

- [20] Sanchari Das, Andrew Kim, and L Jean Camp. “Organizational Security: Implementing a Risk-Reduction-Based Incentivization Model for MFA Adoption”. In: *Proceedings of the International Conference on Financial Cryptography and Data Security*. 2021.
- [21] Sanchari Das, Bingxing Wang, and L Jean Camp. “Mfa is a waste of time! understanding negative connotation towards mfa applications via user generated content”. In: *Proceedings of the Thirteenth International Symposium on Human Aspects of Information Security & Assurance (HAISA 2019)*. 2019.
- [22] Sanchari Das et al. “Towards implementing inclusive authentication technologies for older adults”. In: *Who Are You* (2019).
- [23] Emiliano De Cristofaro et al. “A comparative usability study of two-factor authentication”. In: *arXiv preprint arXiv:1309.5344* (2013).
- [24] Ryan Deng, Weikeng Chen, and Raluca Ada Popa. “N-for-1-Auth: N-wise Decentralized Authentication via One Authentication”. MA thesis. EECS Department, University of California, Berkeley, Dec. 2021. URL: <http://www2.eecs.berkeley.edu/Pubs/TechRpts/2021/EECS-2021-240.html>.
- [25] Periwinkle Doerfler et al. “Evaluating Login Challenges as a Defense Against Account Takeover”. In: *The World Wide Web Conference*. ACM. 2019, pp. 372–382.
- [26] A. Drozhzhin. *SMS-based two-factor authentication is not safe – consider these alternative 2FA methods instead*. Kaspersky Daily. Oct. 2018. URL: <https://usa.kaspersky.com/blog/2fa-practical-guide/16398/>.
- [27] Duo. *How does Duo Push work? What makes it so secure?* URL: [https://help.duo.com/s/article/3252?language=en\\_US](https://help.duo.com/s/article/3252?language=en_US) (visited on 12/07/2022).
- [28] *Duo Privacy Data Sheet*. Aug. 2022. URL: <https://web.archive.org/web/20221004030758/https://trustportal.cisco.com/c/dam/r/ctp/docs/privacydatasheet/security/cisco-duo-privacy-data-sheet.pdf> (visited on 10/03/2022).
- [29] Jonathan Dutson et al. ““Don’t punish all of us”: Measuring User Attitudes about Two-Factor Authentication”. In: *4th European Workshop on Usable Security (EuroUSEC)*. IEEE. 2019.
- [30] Max Eddy. *The Best Two-Factor Authentication App*. 2024. URL: <https://www.nytimes.com/wirecutter/reviews/best-two-factor-authentication-app/> (visited on 04/20/2025).
- [31] S. Egelman et al. “Does my password go up to eleven? The impact of password meters on password selection”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2013, pp. 2379–2388.
- [32] S. Egelman et al. “Please continue to hold”. In: *9th Workshop on the Economics of Information Security*. 2010.

- [33] *Encrypt a Realm - Java SDK — MongoDB Realm*. URL: <https://web.archive.org/web/20220525164833/https://www.mongodb.com/docs/realm/sdk/java/advanced-guides/encryption/> (visited on 05/25/2022).
- [34] Facebook Security. *Introducing Trusted Contacts*. [Online; accessed: 29-September-2021]. May 2013. URL: <https://www.facebook.com/notes/facebook-security/introducing-trusted-contacts/10151362774980766>.
- [35] Facebook Security. *National Cybersecurity Awareness Month Updates*. Oct. 2011. URL: <https://www.facebook.com/notes/facebook-security/national-cybersecurity-awareness-month-updates/10150335022240766> (visited on 09/29/2021).
- [36] Florian M Farke et al. ““You still use the password after all”—Exploring FIDO2 Security Keys in a Small Company”. In: *Sixteenth Symposium on Usable Privacy and Security ({SOUPS} 2020)*. 2020, pp. 19–35.
- [37] FIDO Alliance. *FIDO2: WebAuthn & CTAP*. Sept. 2021. URL: <https://fidoalliance.org/fido2/>.
- [38] *FIDO Alliance and the Path to the Post-Password World*. May 2020. URL: <https://media.fidoalliance.org/wp-content/uploads/2020/05/FIDO-Consumer-Research-Report.pdf> (visited on 06/07/2022).
- [39] Dinei Florencio and Cormac Herley. “A large-scale study of web password habits”. In: *Proceedings of the 16th international conference on World Wide Web*. ACM. 2007, pp. 657–666.
- [40] Dinei Florêncio and Cormac Herley. “Where do security policies come from?” In: *Proceedings of the Sixth Symposium on Usable Privacy and Security*. SOUPS ’10. Redmond, Washington, USA: Association for Computing Machinery, 2010. ISBN: 9781450302647. DOI: 10.1145/1837110.1837124. URL: <https://doi-org.libproxy.berkeley.edu/10.1145/1837110.1837124>.
- [41] Alain Forget, Sonia Chiasson, and Robert Biddle. “Helping users create better passwords: Is this the right approach?” In: *Proceedings of the 3rd Symposium on Usable Privacy and Security*. ACM. 2007, pp. 151–152.
- [42] Nick Frymann et al. “Asynchronous Remote Key Generation: An Analysis of Yubico’s Proposal for W3C WebAuthn”. In: *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 2020, pp. 939–954.
- [43] Eva Gerlitz et al. “Adventures in Recovery Land: Testing the Account Recovery of Popular Websites When the Second Factor is Lost”. In: *Proceedings of the Nineteenth Symposium on Usable Privacy and Security (SOUPS 2023)*. USENIX Association, 2023, pp. 227–243. URL: <https://www.usenix.org/conference/soups2023/presentation/gerlitz> (visited on 04/18/2025).
- [44] Conor Gilsean. *2FA Stats*. 2021. URL: <https://allthingsauth.com/2fastats> (visited on 06/07/2022).

- [45] Conor Gilsenan. *TOTP: (way) more secure than SMS, but more annoying than push*. Apr. 2018. URL: <https://allthingsauth.com/2018/04/05/totp-way-more-secure-than-sms-but-more-annoying-than-push/> (visited on 04/18/2025).
- [46] Conor Gilsenan, Noura Alomar, and Serge Egelman. “On Conducting Systematic Security and Privacy Analyses of TOTP 2FA Apps”. In: *Who Are You?! Adventures in Authentication Workshop*. WAY ’20. Aug. 2020, pp. 1–6.
- [47] Conor Gilsenan et al. “Security and Privacy Failures in Popular 2FA Apps”. In: *32nd USENIX Security Symposium (USENIX Security 23)*. Anaheim, CA: USENIX Association, Aug. 2023, pp. 2079–2096. ISBN: 978-1-939133-37-3. URL: <https://www.usenix.org/conference/usenixsecurity23/presentation/gilsenan>.
- [48] Maximilian Golla et al. “Driving 2FA Adoption at Scale: Optimizing Two-Factor Authentication Notification Design Patterns”. In: *30th {USENIX} Security Symposium ({USENIX} Security 21)*. 2021, pp. 109–126.
- [49] P. Grassi et al. *Digital Identity Guidelines: Authentication and Lifecycle Management*. en. 2017. DOI: <https://doi.org/10.6028/NIST.SP.800-63b>.
- [50] HashiCorp. *Seal/Unseal - HashiCorp Vault Developer Documentation*. URL: <https://developer.hashicorp.com/vault/docs/concepts/seal#shamir-seals> (visited on 12/07/2022).
- [51] D. He, T. Katz, and C. Brand. *Introducing portability of Google Authenticator 2SV codes across Android devices*. URL: <https://web.archive.org/web/20210613033604/https://security.googleblog.com/2020/05/introducing-portability-of-google.html> (visited on 05/26/2022).
- [52] Cormac Herley and Paul Van Oorschot. “A research agenda acknowledging the persistence of passwords”. In: *IEEE Security & Privacy* 10.1 (2011), pp. 28–36.
- [53] *How to Use LastPass Password Manager*. URL: <https://web.archive.org/web/20220606190625/https://www.lastpass.com/how-lastpass-works> (visited on 06/06/2022).
- [54] Nan Hu, Jie Zhang, and Paul A. Pavlou. “Overcoming the J-shaped distribution of product reviews”. In: *Commun. ACM* 52.10 (Oct. 2009), pp. 144–147. ISSN: 0001-0782. DOI: 10.1145/1562764.1562800. URL: <https://doi-org.libproxy.berkeley.edu/10.1145/1562764.1562800>.
- [55] Muhammad Ikram et al. “More Than Just a Random Number Generator! Unveiling the Security and Privacy Risks of Mobile OTP Authenticator Apps”. In: *Proceedings of the 25th International Conference on Web Information Systems Engineering (WISE 2024)*. Vol. 15440. Lecture Notes in Computer Science. Springer, 2024, pp. 177–192. DOI: 10.1007/978-981-96-0576-7\_14. URL: [https://doi.org/10.1007/978-981-96-0576-7\\_14](https://doi.org/10.1007/978-981-96-0576-7_14) (visited on 04/18/2025).
- [56] *Introduction - libsodium*. Mar. 2022. URL: <https://web.archive.org/web/20220531235007/https://doc.libsodium.org/> (visited on 05/31/2022).

- [57] *Introduction to multi-factor authentication (MFA)*. URL: <https://web.archive.org/web/20220607231957/https://help.zoho.com/portal/en/kb/accounts/multi-factor-authentication/articles/mfa-introduction> (visited on 06/07/2022).
- [58] Iulia Ion, Rob Reeder, and Sunny Consolvo. ““...No One Can Hack My Mind”: Comparing Expert and Non-Expert Security Practices”. In: *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*. Ottawa: USENIX Association, 2015, pp. 327–346. ISBN: 978-1-931971-24-9. URL: <https://www.usenix.org/conference/soups2015/proceedings/presentation/ion>.
- [59] Vasu Jakkal. *This World Password Day consider ditching passwords altogether - Microsoft Security Blog*. May 2022. URL: <https://www.microsoft.com/en-us/security/blog/2022/05/05/this-world-password-day-consider-ditching-passwords-together/> (visited on 12/08/2022).
- [60] *Key Uri Format*. URL: <https://github.com/google/google-authenticator/wiki/Key-Uri-Format> (visited on 05/12/2020).
- [61] Keystone. *Why Keystone implemented Shamir Backups*. June 2021. URL: <https://blog.keystone.one/why-keystone-implemented-shamir-backups-71e319f972a6> (visited on 12/07/2022).
- [62] L. L. Kupper and K. B. Hafner. “On Assessing Interrater Agreement for Multiple Attribute Responses”. In: *Biometrics* 45.3 (Sept. 1989), pp. 957–967. DOI: 10.2307/2531695. URL: <https://doi.org/10.2307/2531695> (visited on 04/13/2025).
- [63] LastPass. *LastPass Technical Whitepaper*. Mar. 2021. URL: <https://web.archive.org/web/20250428191723/https://assets.cdngetgo.com/da/ce/d211c1074dea84e06cad6f2c8b8e/lastpass-technical-whitepaper.pdf> (visited on 04/28/2025).
- [64] K. Lee and A. Narayanan. “Security and Privacy Risks of Number Recycling at Mobile Carriers in the United States”. In: *Symposium on Electronic Crime Research (APWG eCrime)*. IEEE. IEEE, Dec. 2021. ISBN: 978-1-6654-8029-1. DOI: 10.1109/eCrime54498.2021.9738792. URL: <https://recyclednumbers.cs.princeton.edu/assets/recycled-numbers-latest.pdf>.
- [65] K. Lee et al. “An Empirical Study of Wireless Carrier Authentication for SIM Swaps”. In: *16th Symposium on Usable Privacy and Security (SOUPS 2020)*. USENIX Association, Aug. 2020, pp. 61–79. ISBN: 978-1-939133-16-8. URL: <https://www.usenix.org/conference/soups2020/presentation/lee>.
- [66] Yue Li, Haining Wang, and Kun Sun. “Email as a master key: Analyzing account recovery in the wild”. In: *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE. 2018, pp. 1646–1654.
- [67] Sanam Ghorbani Lyastani et al. “Is FIDO2 the kingslayer of user authentication? A comparative usability study of FIDO2 passwordless authentication”. In: *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2020, pp. 268–285.

- [68] D. M'Raihi et al. *TOTP: Time-Based One-Time Password Algorithm*. [Online; accessed: 02-Oct-2019]. May 2011. URL: <https://tools.ietf.org/html/rfc6238>.
- [69] Robbie MacGregor. "Evaluating the Android Security Key Scheme: An Early Usability, Deployability, Security Evaluation with Comparative Analysis". In: *Who Are You* (2019), pp. 1–6.
- [70] AbdelKarim Mardini and Guemmy Kim. *Making sign-in safer and more convenient*. Oct. 2021. URL: <https://web.archive.org/web/20220607220259/https://blog.google/technology/safety-security/making-sign-safer-and-more-convenient/> (visited on 06/07/2022).
- [71] W. Melicher et al. "Usability and Security of Text Passwords on Mobile Devices". In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI'16: CHI Conference on Human Factors in Computing Systems. San Jose California USA: ACM, May 7, 2016, pp. 527–539. ISBN: 978-1-4503-3362-7. DOI: 10.1145/2858036.2858384. URL: <https://dl.acm.org/doi/10.1145/2858036.2858384> (visited on 06/08/2022).
- [72] K. Moriarty, B. Kaliski, and A. Rusch. *PKCS #5: Password-Based Cryptography Specification Version 2.1*. RFC 8018. Jan. 2017. DOI: 10.17487/RFC8018. URL: <https://www.rfc-editor.org/info/rfc8018>.
- [73] K. Moriarty et al. *PKCS #12: Personal Information Exchange Syntax v1.1*. RFC 7292. July 2014. DOI: 10.17487/RFC7292. URL: <https://www.rfc-editor.org/info/rfc7292>.
- [74] Jakob Nixdorf. *andOTP Deprecation Announcement*. URL: <https://web.archive.org/web/20221004024933/https://forum.xda-developers.com/t/unmaintained-app-4-4-open-source-andotp-open-source-two-factor-authentication-for-android.3636993/page-6%5C#js-post-87021655> (visited on 03/10/2022).
- [75] Matt Oksa. *Shamir Backup — A Better Way to Secure Your Keys*. Aug. 2019. URL: <https://blog.trezor.io/shamir-backup-the-revolution-of-private-keys-backup-is-here-858687ed7fe7> (visited on 12/07/2022).
- [76] *Our Zero-Knowledge Security Model*. URL: <https://web.archive.org/web/20220606190618/https://www.lastpass.com/security/zero-knowledge-security> (visited on 06/06/2022).
- [77] Kentrell Owens et al. "User Perceptions of the Usability and Security of Smartphones as FIDO2 Roaming Authenticators". In: *Seventeenth Symposium on Usable Privacy and Security ({SOUPS} 2021)*. 2021, pp. 57–76.
- [78] Can Ozkan and Kemal Bicakci. "Security Analysis of Mobile Authenticator Applications". In: *2020 International Conference on Information Security and Cryptology (ISCTURKEY)*. Ankara, Turkey: IEEE, Dec. 3, 2020, pp. 18–30. ISBN: 978-1-66541-863-8. DOI: 10.1109/ISCTURKEY51113.2020.9308020. URL: <https://ieeexplore.ieee.org/document/9308020/> (visited on 09/08/2021).



- [79] *Password Storage - OWASP Cheat Sheet Series*. URL: [https://web.archive.org/web/20220530233607/https://cheatsheetseries.owasp.org/cheatsheets/Password\\_Storage\\_Cheat\\_Sheet.html](https://web.archive.org/web/20220530233607/https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html) (visited on 05/30/2022).
- [80] C. Percival and S. Josefsson. *The script Password-Based Key Derivation Function*. RFC 7914. RFC Editor, Aug. 2016.
- [81] Thanasis Petsas et al. “Two-factor authentication: is the world ready? Quantifying 2FA adoption”. In: *Proceedings of the eighth european workshop on system security*. 2015, pp. 1–7.
- [82] P. Polleit and M. Spreitzenbarth. “Defeating the Secrets of OTP Apps”. In: *11th International Conference on IT Security Incident Management & IT Forensics (IMF)*. Hamburg: IEEE, May 2018, pp. 76–88. ISBN: 978-1-5386-6632-6. DOI: 10.1109/IMF.2018.00013. URL: <https://ieeexplore.ieee.org/document/8514834/> (visited on 06/06/2022).
- [83] *Psychology of Passwords: How Password Hygiene Reduces Your Password Security Risk*. 2020. URL: <https://web.archive.org/web/20220607215159/https://www.lastpass.com/resources/ebook/psychology-of-passwords-2020> (visited on 06/07/2022).
- [84] Ariel Rabkin. “Personal Knowledge Questions for Fallback Authentication: Security Questions in the Era of Facebook”. In: *Proceedings of the 4th Symposium on Usable Privacy and Security*. SOUPS ’08. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 2008, pp. 13–23. ISBN: 9781605582764. DOI: 10.1145/1408664.1408667. URL: <https://doi.org/10.1145/1408664.1408667>.
- [85] Simone Raponi and Roberto Di Pietro. “A longitudinal study on web-sites password management (in) security: Evidence and remedies”. In: *IEEE Access* 8 (2020), pp. 52075–52090.
- [86] J. Reardon et al. “50 Ways to Leak Your Data: An Exploration of Apps’ Circumvention of the Android Permissions System”. In: *Proceedings of the 28th USENIX Security Symposium*. 2019, pp. 603–620.
- [87] E. M. Redmiles and E. Hargittai. “New phone, who dis? Modeling millennials’ backup behavior”. In: *ACM Transactions on the Web (TWEB)* 13.1 (2018), pp. 1–14.
- [88] E. M. Redmiles, S. Kross, and M. L. Mazurek. “How I Learned to Be Secure: A Census-Representative Survey of Security Advice Sources and Behavior”. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. CCS ’16. Vienna, Austria: ACM, 2016, pp. 666–677. ISBN: 978-1-4503-4139-4. DOI: 10.1145/2976749.2978307. URL: <http://doi.acm.org/10.1145/2976749.2978307>.
- [89] Elissa M Redmiles, Everest Liu, and Michelle L Mazurek. “You Want Me To Do What? A Design Study of Two-Factor Authentication Messages.” In: *SOUPS*. 2017.

- [90] Elissa M. Redmiles, Sean Kross, and Michelle L. Mazurek. “How I Learned to Be Secure: A Census-Representative Survey of Security Advice Sources and Behavior”. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. CCS ’16. Vienna, Austria: ACM, 2016, pp. 666–677. ISBN: 978-1-4503-4139-4. DOI: 10.1145/2976749.2978307. URL: <http://doi.acm.org/10.1145/2976749.2978307>.
- [91] Robert Reeder and Stuart Schechter. “When the password doesn’t work: Secondary authentication for websites”. In: *IEEE Security & Privacy* 9.2 (2011), pp. 43–49.
- [92] Ken Reese et al. “A usability study of five two-factor authentication methods”. In: *Fifteenth Symposium on Usable Privacy and Security ({SOUPS} 2019)*. 2019, pp. 357–370.
- [93] Irwin Reyes et al. ““Won’t Somebody Think of the Children?” Examining COPPA Compliance at Scale”. In: *Proceedings on Privacy Enhancing Technologies* 2018.3 (2018), pp. 63–83. DOI: doi:10.1515/popets-2018-0021. URL: <https://doi.org/10.1515/popets-2018-0021>.
- [94] Joshua Reynolds et al. “A tale of two studies: The best and worst of yubikey usability”. In: *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2018, pp. 872–888.
- [95] Joshua Reynolds et al. “Empirical Measurement of Systemic 2FA Usability”. In: *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, Aug. 2020, pp. 127–143. ISBN: 978-1-939133-17-5. URL: <https://www.usenix.org/conference/usenixsecurity20/presentation/reynolds>.
- [96] Salesforce. *Back Up Your Connected Accounts in the Salesforce Authenticator Mobile App*. URL: [https://help.salesforce.com/s/articleView?id=sf.salesforce\\_authenticator\\_backup.htm&type=5](https://help.salesforce.com/s/articleView?id=sf.salesforce_authenticator_backup.htm&type=5) (visited on 10/03/2022).
- [97] Salesforce. *Restore Connected Accounts in the Salesforce Authenticator Mobile App*. URL: [https://help.salesforce.com/s/articleView?id=sf.salesforce\\_authenticator\\_restore\\_from\\_backup.htm&type=5](https://help.salesforce.com/s/articleView?id=sf.salesforce_authenticator_restore_from_backup.htm&type=5) (visited on 10/03/2022).
- [98] Stuart Schechter, A.J. Bernheim Brush, and Serge Egelman. “It’s No Secret. Measuring the Security and Reliability of Authentication via”. In: *Proceedings of the 2009 IEEE Symposium on Security and Privacy*. Los Alamitos, CA, USA: IEEE Computer Society, 2009, pp. 375–390. DOI: <http://doi.ieeecomputersociety.org/10.1109/SP.2009.11>.
- [99] Stuart Schechter, Serge Egelman, and Robert W. Reeder. “It’s not what you know, but who you know: a social approach to last-resort authentication”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’09. Boston, MA, USA: ACM, 2009, pp. 1983–1992. ISBN: 978-1-60558-246-7. DOI: 10.1145/1518701.1519003. URL: <http://doi.acm.org/10.1145/1518701.1519003>.

- [100] Adi Shamir. “How to share a secret”. In: *Communications of the ACM* 22.11 (1979), pp. 612–613.
- [101] Richard Shay et al. “Encountering stronger password requirements: user attitudes and behaviors”. In: *Proceedings of the Sixth Symposium on Usable Privacy and Security*. ACM. 2010, p. 2.
- [102] Jordan Shropshire and Philip Menard. “A new approach to mobile device authentication”. In: *Proceedings of the 10th annual Workshop on Information Security and Privacy*. 2015.
- [103] *SLIP-0039 : Shamir’s Secret-Sharing for Mnemonic Codes*. URL: <https://github.com/satoshilabs/slips/blob/master/slip-0039.md> (visited on 12/07/2022).
- [104] Keira Stevens. *Hashes, Salts, and Rainbow Tables: Confessions of a Password Cracker*. Dark Reading. May 2021. URL: <https://www.darkreading.com/application-security/hashes-salts-and-rainbow-tables-confessions-of-a-password-cracker/a/d-id/1340928>.
- [105] The Linux Foundation. *Case Study: Let’s Encrypt*. 2020. URL: <https://www.linuxfoundation.org/resources/case-studies/lets-encrypt> (visited on 04/20/2025).
- [106] Kurt Thomas et al. “Data breaches, phishing, or malware?: Understanding the risks of stolen credentials”. In: *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*. ACM. 2017, pp. 1421–1434.
- [107] B. Ur et al. “Design and Evaluation of a Data-Driven Password Meter”. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI ’17: CHI Conference on Human Factors in Computing Systems. Denver Colorado USA: ACM, May 2, 2017, pp. 3775–3786. ISBN: 978-1-4503-4655-9. DOI: 10.1145/3025453.3026050. URL: <https://dl.acm.org/doi/10.1145/3025453.3026050> (visited on 06/08/2022).
- [108] Emily A. Vogels and Monica Anderson. *Americans and Digital Knowledge*. Oct. 2019. URL: <https://www.pewresearch.org/internet/2019/10/09/americans-and-digital-knowledge/> (visited on 04/17/2025).
- [109] Emanuel von Zezschwitz, Alexander De Luca, and Heinrich Hussmann. “Honey, I Shrunk the Keys: Influences of Mobile Devices on Password Composition and Authentication Performance”. In: *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*. NordiCHI ’14: The 8th Nordic Conference on Human-Computer Interaction. Helsinki Finland: ACM, Oct. 26, 2014, pp. 461–470. ISBN: 978-1-4503-2542-4. DOI: 10.1145/2639189.2639218. URL: <https://dl.acm.org/doi/10.1145/2639189.2639218> (visited on 06/08/2022).
- [110] Jake Weidman and Jens Grossklags. “I like it, but i hate it: Employee perceptions towards an institutional transition to byod second-factor authentication”. In: *Proceedings of the 33rd Annual Computer Security Applications Conference*. 2017, pp. 212–224.

- [111] Alex Weinert. *How it works: Backup and restore for Microsoft Authenticator*. URL: <https://web.archive.org/web/20220630215634/https://techcommunity.microsoft.com/t5/microsoft-entra-azure-ad-blog/how-it-works-backup-and-restore-for-microsoft-authenticator/ba-p/1006678> (visited on 06/30/2022).
- [112] Zack Whittaker. *How to keep your Twitter secure without giving Elon Musk any money*. Feb. 2023. URL: <https://techcrunch.com/2023/02/18/how-to-keep-twitter-secure-two-factor/> (visited on 04/20/2025).
- [113] Primal Wijesekera et al. “Android Permissions Remystified: A Field Study on Contextual Integrity”. In: *24th USENIX Security Symposium (USENIX Security 15)*. Washington, D.C.: USENIX Association, Aug. 2015, pp. 499–514. ISBN: 978-1-931971-232. URL: <https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/wijesekera>.
- [114] Primal Wijesekera et al. “The Feasability of Dynamically Granted Permissions: Aligning Mobile Privacy with User Preferences”. In: *Proceedings of the 2017 IEEE Symposium on Security and Privacy*. Oakland '17. IEEE Computer Society, 2017.
- [115] Diego Zavala et al. *Android Developers Blog: Bringing passkeys to Android & Chrome*. Oct. 2022. URL: <https://android-developers.googleblog.com/2022/10/bringing-passkeys-to-android-and-chrome.html> (visited on 12/08/2022).
- [116] Zoho. *Secure non-Zoho accounts using OneAuth’s OTP authenticator*. URL: [https://web.archive.org/web/20221003160327/https://help.zoho.com/portal/en/kb/accounts/oneauth/v2/articles/otp-authenticator%5C#Back\\_up\\_and\\_restore\\_OTP\\_secrets](https://web.archive.org/web/20221003160327/https://help.zoho.com/portal/en/kb/accounts/oneauth/v2/articles/otp-authenticator%5C#Back_up_and_restore_OTP_secrets) (visited on 10/03/2022).

# Appendix A

## Supplemental Materials for Chapter 4

### A.1 Full App Names and Versions

See Table A.1 for a mapping of the abbreviated app name used throughout the paper to the full app name, Google Play Store URL, and the APK version that was analyzed in this work. Note that the URL includes the APK id as a query parameter. Readers can access the specific APK versions analyzed in this work using a mirror service, such as [apkmirror.com](https://apkmirror.com).

#	Abbreviated Name	Full Name	Google Play Store URL	Version
1	Google Authenticator	Google Authenticator	<a href="https://play.google.com/store/apps/details?id=com.google.android.apps.authenticator2">https://play.google.com/store/apps/details?id=com.google.android.apps.authenticator2</a>	v5.10
2	Microsoft Authenticator	Microsoft Authenticator	<a href="https://play.google.com/store/apps/details?id=com.azure.authenticator">https://play.google.com/store/apps/details?id=com.azure.authenticator</a>	v6.2204.2757
3	Duo Mobile	Duo Mobile	<a href="https://play.google.com/store/apps/details?id=com.duosecurity.duomobile">https://play.google.com/store/apps/details?id=com.duosecurity.duomobile</a>	v4.15.0
4	Twilio Authy	Twilio Authy 2-Factor Authentication	<a href="https://play.google.com/store/apps/details?id=com.authy.authy">https://play.google.com/store/apps/details?id=com.authy.authy</a>	v24.8.5
5	Latch	Latch	<a href="https://play.google.com/store/apps/details?id=com.elevenpaths.android.latch">https://play.google.com/store/apps/details?id=com.elevenpaths.android.latch</a>	v2.2.4
6	LastPass Authenticator	LastPass Authenticator	<a href="https://play.google.com/store/apps/details?id=com.lastpass.authenticator">https://play.google.com/store/apps/details?id=com.lastpass.authenticator</a>	v2.5.0
7	2FAS	2FA Authenticator (2FAS)	<a href="https://play.google.com/store/apps/details?id=com.twofasapp">https://play.google.com/store/apps/details?id=com.twofasapp</a>	v3.11.0
8	Yandex.Key	Yandex.Key	<a href="https://play.google.com/store/apps/details?id=ru.yandex.key">https://play.google.com/store/apps/details?id=ru.yandex.key</a>	v2.7.0
9	FreeOTP Authenticator	FreeOTP Authenticator	<a href="https://play.google.com/store/apps/details?id=org.fedorahosted.freeotp">https://play.google.com/store/apps/details?id=org.fedorahosted.freeotp</a>	v1.5
10	Authenticator	Authenticator	<a href="https://play.google.com/store/apps/details?id=com.pixplicity.auth">https://play.google.com/store/apps/details?id=com.pixplicity.auth</a>	v1.0.6
11	Salesforce Authenticator	Salesforce Authenticator	<a href="https://play.google.com/store/apps/details?id=com.salesforce.authenticator">https://play.google.com/store/apps/details?id=com.salesforce.authenticator</a>	v3.8.5
12	Code Generator	Code Generator	<a href="https://play.google.com/store/apps/details?id=net.codemonkey.otpgeneratorapp">https://play.google.com/store/apps/details?id=net.codemonkey.otpgeneratorapp</a>	v6.1
13	TOTP Authenticator	TOTP Authenticator – 2FA with Cloud Sync & Widget	<a href="https://play.google.com/store/apps/details?id=com.authenticator.authservice2">https://play.google.com/store/apps/details?id=com.authenticator.authservice2</a>	v1.89
14	Aegis Authenticator	Aegis Authenticator - Two Factor (2FA) app	<a href="https://play.google.com/store/apps/details?id=com.beemdevelopment.aegis">https://play.google.com/store/apps/details?id=com.beemdevelopment.aegis</a>	v2.0.3
15	Auth0 Guardian	Auth0 Guardian	<a href="https://play.google.com/store/apps/details?id=com.auth0.guardian">https://play.google.com/store/apps/details?id=com.auth0.guardian</a>	v1.5.3
16	App Authenticator	Authenticator : App Authenticator	<a href="https://play.google.com/store/apps/details?id=authentic.your.app.authenticator">https://play.google.com/store/apps/details?id=authentic.your.app.authenticator</a>	v1.5
17	andOTP	andOTP - Android OTP Authenticator	<a href="https://play.google.com/store/apps/details?id=org.shadowice.flocke.andotp">https://play.google.com/store/apps/details?id=org.shadowice.flocke.andotp</a>	v0.9.0.1-play
18	Zoho OneAuth	Zoho OneAuth - Authenticator	<a href="https://play.google.com/store/apps/details?id=com.zoho.accounts.oneauth">https://play.google.com/store/apps/details?id=com.zoho.accounts.oneauth</a>	v2.1.0.5
19	Authenticator Pro	Authenticator Pro — offline 2FA Two Factor app	<a href="https://play.google.com/store/apps/details?id=me.jmh.authenticatorpro">https://play.google.com/store/apps/details?id=me.jmh.authenticatorpro</a>	v1.15.10
20	SAASPASS	SAASPASS Authenticator 2FA App & Password Manager	<a href="https://play.google.com/store/apps/details?id=com.solidpass.saaspass">https://play.google.com/store/apps/details?id=com.solidpass.saaspass</a>	v2.2.28
21	Authentic Password	Authenticator : Authentic Password	<a href="https://play.google.com/store/apps/details?id=authentic.password.authenticator.pro">https://play.google.com/store/apps/details?id=authentic.password.authenticator.pro</a>	v1.3
22	Mobile Authenticator	Authenticator : Mobile Authenticator App	<a href="https://play.google.com/store/apps/details?id=authenticator.mobile.authenticator">https://play.google.com/store/apps/details?id=authenticator.mobile.authenticator</a>	v1.7

Table A.1: Mapping of abbreviated app names used throughout the paper to identifying metadata and version information.

## A.2 Base32 decryption heuristic

*Given the ciphertext of the encrypted TOTP secret, what is the probability that a single password guess will generate a plaintext output that is valid Base32?* An ASCII character has  $2^8 = 256$  possible bit permutations and Base32 allows 32 valid characters (A-Z and 2-7). The probability that an L-length ASCII string is valid Base32 is:

$$= P(\text{single byte is valid Base32})^L = (32/256)^L = 0.125^L$$

The probability that a single password guess for a 32 byte TOTP secret ( $L = 32$ ), which is a common length used in industry, will generate valid Base32 is:  $= 0.125^{32} \approx 1.26 * 10^{-29}$ . With a very high probability, this heuristic will accurately verify whether the user entered the correct recovery password because it is extremely unlikely that the decryption process will result in plaintext that is valid Base32 format if the encryption key is derived from an incorrect password.

## A.3 Google Play Search Terms

The following comma separated list contains the 109 final search terms. The 32 core search terms are marked with an \*. See Section 4.1.1 for more details.

2 factor\*, 2 factor authentication\*, 2 factor authentication app, 2 factor authentication app facebook, 2 factor authentication app free, 2 factor authentication discord, 2 step \*, 2 step verification app, 2 step verification app walmart, 2 step verification bypass, 2 step verification sony, 2-factor\*, 2-factor authentication\*, 2-factor authentication app, 2-factor authentication code, 2-step verification\*, 2-step verification app, 2fa\*, 2fa app, 2fa authenticator, 2fa authenticator app, 2fa fortnite, 2face, 2sv\*, 2sv authenticator, 2sv authenticator app, 2sv authenticator app sony, 2sv2, authenticator\*, authenticator app fortnite, authenticator app microsoft, authenticator blizzard, authenticator google, code generator\*, code generator app, code generator facebook, code generator for facebook, code generator for facebook login, duo multi factor authentication, mfa\*, mfa app, mfa authenticator, mfa boston, mfa mobile app, multi factor\*, multi factor authentication\*, multi factor authentication app, multi factor authenticator, multi factor authenticator app, multi-factor\*, multi-factor authentication\*, multi-factor authentication (mfa), multi-factor authentication app, multi-factor authentication application, multi-factor authentication mobile app, multifactor\*, multifactor authentication\*, multifactor authentication for office 365, multifactor authentication, multifactorial authentication, one time password\*, one time password (otp), one time password app lock, one time password authenticator, one time password ff14, one time password ffxiv, one time password square enix, one-time password\*, otp\*, otp app, otp authenticator, otp authenticator samsung, otpauth, setup multi factor authentication, tfa\*, tfa app, tface, tfacebook, tfanslate, time based one time password\*, time based one-time password\*, time-based one time password\*, time-based one-time password\*, totp\*, totp app, totp authenticator, totp

authenticator app, totp-compliant authentication software, two factor\*, two factor authentication\*, two factor authentication app, two factor authentication app discord, two factor authentication app facebook, two factor authentication app fortnite, two step verification\*, two step verification app, two step verification discord, two step verification facebook, two step verification walmart, two step verification whatsapp, two-factor\*, two-factor authentication\*, two-factor authentication (2fa), two-factor authentication app, two-factor authentication code, two-factor authenticator, two-factor authenticator app, two-step verification\*, two-step verification token service

## A.4 Network Traffic Snippets

This appendix contains snippets of network traffic for each app that sent personal or sensitive information remotely to the backup service or other third parties. Each snippet has been edited for brevity and to highlight the relevant data.

### A.4.1 Yandex.Key (ru.yandex.key@2.7.0)

The app encrypts the TOTP backup using a symmetric key derived from a user provided password. The traffic snippet below highlights that both the ciphertext and the user password (Password123!) are sent to Yandex servers, which gives Yandex the capability to decrypt the TOTP backup and read its contents, including the TOTP secret, label, and issuer.

```
===== BEGIN REQUEST =====
POST registrator.mobile.yandex.net/1/validation/password/?consumer=dev&
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

password=Password123!
===== END REQUEST =====
===== BEGIN REQUEST =====
POST registrator.mobile.yandex.net/1/bundle/yakey_backup/upload/?
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

number=+12029908424
backup=S0hRm9170EvA9L44YDLIKBBIXK8GoMp04Gm2YDNoSiXiP_nR-rsuUi608lt1ZiwUgnqQg
↪ KyQHi2CVs4V-wZKkpJCer6puvmGMy89jZIRXgHCH-6_WJPNVyZfWyRoXXXZPNeL_s9HspqAT
↪ GZiiJGaKyEths8FsJPN2WX64k2FaiNq2PbX6x2_rsv5YTvysQM0kGImpus3VQhA6FL1I5nvR
↪ TazoZ9j-Lh0es69ysZgpyLp4J9AwY_okUsVbu0v1QdKklU
device_name=Google Pixel 3a
===== END REQUEST =====
```



### A.4.2 Zoho OneAuth - Authenticator (com.zoho.accounts.oneauth)

The app encrypts the TOTP backup using a symmetric key derived from a user provided password. The traffic snippet below highlights that both the ciphertext and the user password (`rRmhyojrhMyAFFD2_sEdXwiyJfTexRBr`) are sent to Zoho servers, which gives Zoho the capability to decrypt the TOTP backup and read its contents, including the TOTP secret, label, and issuer. Also sent to Zoho are the plaintext values of the issuer (`Research Lab`) and label (e.g., `example-email-1@example.com`).

===== BEGIN REQUEST =====

POST accounts.zoho.com/api/v1/account/self/user/self/passphrase?device\_token

↪ ...&nonce=...&

Content-Type: application/json; charset=UTF-8

```
{
  "passphrase": {
    "pass_phrase": "rRmhyojrhMyAFFD2_sEdXwiyJfTexRBr",
    "private_key": "...",
    "public_key": "..."
  }
}
```

===== END REQUEST =====

===== BEGIN REQUEST =====

POST accounts.zoho.com/api/v1/account/self/user/self/tpsecret?device\_token=.

↪ ..&nonce=...&

Content-Type: application/json; charset=UTF-8

```
{
  "tpsecret_all": [
    {
      "app_name": "Research Lab",
      "app_secret":
        ↪ "GLUEKWXJQCWKMIORCAB3HJLGVO4QKD7FCBQRKYKT2UFYWOAZ054DYQLXGLNL5FVY
        ↪ OBWJOMB7XTVKZLGH2S2IRPDB6YUT6RXDYMNMX6M4ZIGLCMTGL2HDHWL2GWHZM66",
      "label": "example-email-1@example.com"
    },
    {
      "app_name": "Research Lab",
```

```
"app_secret":  
  ↪ "GLUEKWXJQCWKMIORCAB3HJLGVO4QKD7FCBQRKYKT2UFYWOAZO54DYQPLXGLNL5FVY  
  ↪ OBWJOMB7XTVLU2INMHOYV7LSYJOWPFHE4LPEYOM4ZIGLCMTGL2HDHWL2GWHZM66",  
  "label": "example-email-2@example.com"  
}  
]  
}  
  
===== END REQUEST =====
```

### A.4.3 Twilio Authy 2-Factor Authentication (com.authy.authy)

The traffic snippet below highlights that values of the issuer (Research Lab) and label (e.g., example-email-1@example.com) TOTP fields are sent to Authy servers in plaintext.

```
===== BEGIN REQUEST =====
```

```
POST api.authy.com/json/users/597149415/authenticator_tokens/update
```

```
Content-Type: application/x-www-form-urlencoded
```

```
token_id=1652749577
```

```
account_type=authenticator
```

```
encrypted_seed=xqyRw2btXbd0hAi2YDsQ+4uMbtZoRC4cbvElfl8vKXW0UTubsU3RHM1ZXk6we
```

```
↪ SzD57YF40DKXRLVgus7p5cMXg==
```

```
digits=6
```

```
issuer=Research Lab
```

```
name=Research Lab: example-email-1@example.com
```

```
salt=heXF25paXzbduhVaY3aZPd10CfYLUdA3
```

```
password_timestamp=1652749574
```

```
original_name=Research Lab:example-email-1@example.com
```

```
logo=Research Lab
```

```
===== END REQUEST =====
```

#### A.4.4 Latch (com.elevenpaths.android.latch@v2.2.4)

The traffic snippet below highlights that all TOTP fields, including secret (JBSWY3DPREEQFA3DBNFXHIZLYOQQHGXZLDOJSXIIDOOVWWEZLSEAYQ), label (e.g., example-email-1@example.com), and issuer (Research Lab), are sent to the Latch servers in plaintext.

```
===== BEGIN REQUEST =====
POST latch.elevenpaths.com/control/1.8/totp?
Content-Type: application/x-www-form-urlencoded

period=30
accountName=example-email-1@example.com
name=Research Lab
digits=6
secret=JBSWY3DPREEQFA3DBNFXHIZLYOQQHGXZLDOJSXIIDOOVWWEZLSEAYQ
algorithm=SHA1
===== END REQUEST =====

% ===== BEGIN REQUEST =====
% POST latch.elevenpaths.com/control/1.8/totp?
% Content-Type: application/x-www-form-urlencoded

% period=30
% accountName=example-email-2@example.com
% name=Research Lab
% digits=6
% secret=JBSWY3DPREEQFA3DBNFXHIZLYOQQHGXZLDOJSXIIDOOVWWEZLSEAZA
% algorithm=SHA1
% ===== END REQUEST =====
```

### A.4.5 SAASSPASS Authenticator 2FA App & Password Manager (com.solidpass.saasspass@v2.2.28)

The traffic snippet below highlights that all TOTP fields, including secret (e.g., JBSWY3DP EEQFA3DBNFXHIZLYOQQHGZLDOJSXIIDOOVWWEZLSEAZA), label (e.g., example-email-1@example.com), and issuer (Research Lab), are sent to the SAASSPASS servers in plaintext using the XMPP protocol.

```
===== BEGIN REQUEST to 104.154.49.147:0 (backing up) =====
<message id='6' to='saasspass@saasspass.com'
  → type='chat'><body>{&quot;auth&quot;;:&quot;clientId&quot;;:&quot;xyht1h9k
  → 0fnu4ajqqr3luk2g6c9nyug5&quot;;:&quot;mac&quot;;:&quot;ZQAAJyxwbskA1HzF/wR
  → 2UEikpaDMpjcI3CZENq0jGHvCg6tFhYHjeeS4jwP+p2544K/9EqCTHRokhebXENL6YCUfrDX
  → y&quot;;}&quot;;data&quot;;:&quot;accountType&quot;;:&quot;AUTHENTICATOR&quot;
  → uot;;:&quot;appName&quot;;:&quot;Research
  → Lab&quot;;:&quot;key&quot;;:&quot;otpaath://totp/Research
  → Lab:example-email-2@example.com?secret\u003dJBSWY3DP EEQFA3DBNFXHIZLYOQQH
  → GZLDOJSXIIDOOVWWEZLSEAZA\u0026issuer\u003dResearch
  → Lab\u0026digits\u003d6\u0026algorithm\u003dSHA1&quot;;:&quot;serviceUrl&quot;
  → uot;;:&quot;;:&quot;;:&quot;ssoEnabled&quot;;:&quot;false,&quot;username&quot;;:&quot;
  → t;example-email-2@example.com&quot;;}&quot;;device&quot;;:&quot;dateTime&quot;
  → quot;;:&quot;1650552954057,&quot;knownRevision&quot;;:&quot;8,&quot;language&quot;;:&quot;
  → ot;en&quot;;:&quot;;:&quot;spKnownVersion&quot;;:&quot;2.2.28&quot;;:&quot;;:&quot;spVersio
  → n&quot;;:&quot;2.2.28&quot;;}&quot;;requestId&quot;;:&quot;6,&quot;service&quot;;:&quot;
  → &quot;authenticatorCreator&quot;;:&quot;;:&quot;type&quot;;:&quot;request&quot;;}</
  → body></message>
===== END REQUEST =====
```

```
===== BEGIN RESPONSE from 104.154.49.147:0 while recovering =====
```

```

<message to="xyht1h9k0fnu4ajqqr3luk2g6c9nyug5@saaspass.com/xyht1h9k0fnu4ajqq
→ r3luk2g6c9nyug5" id="3" type="chat"
→ from="saaspass@saaspass.com/76c908d4"><body>{"type":"response","service"
→ :"authenticatorCreator","requestId":3,"result":{"success":true,"message"
→ :null,"code":0},"data":{"revision":4,"revisions":[{"changeset":{"categor
→ y":"email","action":"create","data":{"emailAddress":"example-email-1@exa
→ mple.com","verified":false,"id":6010138944750702191,"allowDelete":true}}
→ ,"revisionId":3},{"changeset":{"category":"authenticator","action":"crea
→ te","data":{"username":"example-email-1@example.com","displayName":null,
→ "id":9218846251831681287,"appName":"Research
→ Lab","key":"otppauth://totp/Research
→ Lab:example-email-1@example.com?secret=JBSWY3DPREEQFA3DBNFXHIZLYOQQHGLD0
→ JSXIID00VWWEZLSEAYQ&issuer=Research
→ Lab&digits=6&algorithm=SHA1"},"ssoEnabled":false,"serviceUrl":nul
→ l,"password":null,"accountType":"AUTHENTICATOR","computerName":null,"com
→ puterType":null,"comType":null,"computerClientId":null,"iconSetVersion":
→ null,"iconSet":null,"storePasswordOnServer":true,"allowAutoLogin":true}}
→ ,"revisionId":4}], "id":9218846251831681287,"username":"example-email-1@e
→ xample.com","displayName":null,"appName":"Research
→ Lab","key":"otppauth://totp/Research
→ Lab:example-email-1@example.com?secret=JBSWY3DPREEQFA3DBNFXHIZLYOQQHGLD0
→ JSXIID00VWWEZLSEAYQ&issuer=Research
→ Lab&digits=6&algorithm=SHA1"},"ssoEnabled":false,"serviceUrl":nul
→ l,"password":null,"accountType":"AUTHENTICATOR","computerName":null,"com
→ puterType":null,"storePasswordOnServer":null,"isMerged":null}}</body></m
→ essage>
===== END RESPONSE =====

```

### A.4.6 2FA Authenticator (2FAS) (com.twofasapp@v3.11.0)

The traffic snippet below highlights that the 2FAS app send all TOTP fields, including secret (e.g., JBSWY3DPREEQFA3DBNFXHIZLYOQQHGZLDOJSXIIDOOVWWEZLSEAYQ), label (e.g., example-email-1@example.com), and issuer (e.g., Research Lab), that have been previously registered in the app to Google Drive in plaintext when the Google Drive backup mechanism is first enabled. After the mechanism is enabled, the app encrypts all TOTP fields when backing up existing and new accounts. This UX bug could be easily remedied.

```
===== BEGIN REQUEST =====
```

```
PUT www.googleapis.com/upload/drive/v3/files
```

```
Content-Type: text/plain
```

```
{ "services": [ { "name": "Research
↳ Lab", "secret": "JBSWY3DPREEQFA3DBNFXHIZLYOQQHGZLDOJSXIIDOOVWWEZLSEAYQ", "up
↳ datedAt": 1653369154872, "type": "Unknown", "otp": { "label": "Research
↳ Lab:example-email-1@example.com", "account": "example-email-1@example.com"
↳ , "issuer": "Research
↳ Lab", "digits": 6, "period": 30, "algorithm": "SHA1", "counter": 1, "tokenType": "
↳ TOTP" }, "order": { "position": 0 } } ], "updatedAt": 1653369154872, "schemaVersion
↳ ": 2, "appVersionCode": 3110000, "appVersionName": "3.11.0", "appOrigin": "andr
↳ oid", "groups": [], "account": "researchlabtotp@gmail.com" }
```

```
===== END REQUEST =====
```

### A.4.7 Salesforce (com.salesforce.authenticator@v3.8.5)

The app encrypts the TOTP backup using a symmetric key derived from a user provided password. The traffic snippet below highlights that both the ciphertext and the user password (0000) was sent to Salesforce servers, which gives Salesforce the capability to decrypt the TOTP backup and read its contents, including the TOTP secret, label, and issuer.

===== BEGIN REQUEST =====

```
POST authenticator-api.salesforce.com/services/verify/v1/authenticators/ae73
  ↪ c008-3089-42ec-961b-2ebb5e0777de/backup?
Content-Type: application/x-www-form-urlencoded
```

```
bundle_secret=4Mkzu8WUgt+N1Y59fGP9BiU4UAdztqW7XTPkRnb0SIc=
passcode_type=numeric_pin
passcode=0000
backup_blob={"encrypted_bundle":"kLJYom7Sh73G3as8K\//a\//xugikHIN1nDrBwW7XML4z
  ↪ H11jHnQdINhAWwNgnAtGgMjux50g\//27DoYVLwPvnFFltLwakb74R1c7qmUGuZgbhyif2v6q
  ↪ aYb71uQvs895krqQD8cBJVTdy18PyU3Dw\//7f2670eAT0zywJEUjmbn2\//vi+\//ZzIDoxuMb
  ↪ sm8A6r\//qhblI53gxQrrcz\//cr5+5Xoi1cNv5tK0ef9kF20N2Us4QM7F0PBN5+GFuaf1FHEa
  ↪ IXAiro85QFWzj5qdzLuzgPf2\//peZIIIGUSlTTnKrzLk8jeG1PRu2tDNF7UJz9AK50p4nevnQ
  ↪ Vjxv3VYvekags\//QGKgXw==", "encrypted_keys": ["SuEOVfZZt9Mwzs0x0CigNLNxnF6d
  ↪ j8eI8Tw0ZzBsxADm4dAKafbQ4erQGbIjgyD6"], "hash_salt": "4+VpmumqWtDmwz1C\//RW
  ↪ w\//Q==", "hash_iterations": 10000, "initialization_vector": "Njadu35JsKiP9zx
  ↪ Eb9go2g=="}
```

===== END REQUEST =====



### A.4.8 Duo Mobile (com.duosecurity.duomobile@v4.15.0)

The traffic snippet below highlights that the Duo Mobile app includes the TOTP label (e.g., `example-email-1@example.com`) in plaintext in the TOTP backup and sometimes includes the TOTP issuer (e.g., `GitHub`) in plaintext as well. If the TOTP issuer is one of the values on a hard coded list, then the app includes the TOTP issuer in the backup; otherwise, it marks the TOTP issuer as “custom” and **does not** backup the TOTP issuer.

===== BEGIN REQUEST =====

```
PUT www.googleapis.com/upload/drive/v3/files/1w81f4K6u401DVB9H1WjmU_a3K7qQrV
↪ ByivXmA53djAHgXwX00g?
Content-Type: application/json
```

```
[
  {
    "version": 1.0,
    "accountType": "OtpAccount",
    "name": "example-email-1@example.com",
    "logoUri": "file:///data/user/0/com.duosecurity.duomobile/files/duokit/1
↪ ogos/f5bfb98d-12f3-4602-b0a0-42e537deae98.png",
    "pkey": "f5bfb98d-12f3-4602-b0a0-42e537deae98",
    "serviceTypeLabelIsCustom": true
  },
  {
    "version": 1.0,
    "accountType": "OtpAccount",
    "logoUri":
↪ "android.resource://com.duosecurity.duomobile/drawable/ic_github",
    "pkey": "42f1e517-f30e-46c8-bbfe-2c6d28cbc67a",
    "serviceTypeLabel": "GitHub",
    "serviceTypeLabelIsCustom": false
  }
]
```

===== END REQUEST =====

### A.4.9 TOTP Authenticator (com.authenticator.authservice2@v1.89)

TOTP authenticator uses a hard-coded, static string (`TotpAuthenticator`) to generate the encryption key, resulting in the same key being used for every piece of information. This, along with a static initialization vector results in no randomness being added to the cryptographic process, meaning that repeated pieces of information can be trivially detected and read.

```
===== BEGIN REQUEST =====
```

```
PUT www.googleapis.com/upload/drive/v3/files?
```

```
Content-Type: text/plain
```

```
kINkyu035fB5RhSlPwiFGJspdlORIjgKxXG/aCLcnrUPDYvDTZWxj0daLko7hKVx2q0cdGFWjE+x
→ /+LaIAagsdrMsJp7HjL+LvQkuKtKGyKME1sVkPCYwat1Y1mwr28zlGTSW/vl6Zku+5o2k3/a
→ ls9080LowQ0/stmta3G+CEcI4kQSfDnBcUctdhhoWiDRigyHrpLT2IPPMQu+P3qFLGfloWM1
→ 3rULBTKJZm82Qfp16/SvWxn1BFeUFSxfFB6frAd4ajvU+oUH6rMuacczayNe0JjupiJcKoVg
→ X7DUnyrC0yaXJxUQzRJbEFwHzlok7nFEULgEMoxYTw11N1JZsfNtHcq4zGKVGgBSTBpxnEbN
→ A7QLA/QZ3ggfroz3+29NZ4BVKkc+a/7QQg9k8wi1kmDXPpHu0s8n05dQZANCCiWjzJzv+ktQ
→ x5GDmX+BGvTPQjjju0jCY+coasQKw2a/DGn7CQxtsR5XVIS00Saoud+SVPkBEmmnpKBSHEI1
→ jgvAkbjz7rX2QvIj5iHeZ5qA6Uz687LgONRdc2aCAWFY7TnqMgfDY404xaJS7FB+XpdTk0+1
→ zC0aG18e8x7LxMpjaahQVq2SkItG1MM8+Y1a1Qcd3VBbJqT/qaAMJtF5mVPyVT2mloJBnHbC
→ N4u2bGzh0iSCnTSeLbcWF53BRPz+NpadGw5tWSokDQnMFTj81NQ0Kb2cR1RrnnR5q1IdHSi3
→ 7ZGhaQ4Ipfrd+1kx9yRPJ+DJyX10HD1Ko09t85v9m1dLHXfNMDzLPP7YwUFrtbeHEvhA3Ku7
→ hFkp9tsLvDqAsrhEbe8F6v0s/5LvTmoORDI/mze6b0bTgTSjDDz5eao433DjekFlZnjUIi0
→ QpgYyULKioZKPAFN8COZZzs5dy1lJkVNfVc6JcQQNJU/HJVFyzDe+U21/kyzPok0pMXZtqV9
→ 4vAujRgTUowBeFbbhBvVOGmUDEVHbeVDXJt7ieZa77Q1VfgaPRRyJ2E74ExhL4M=
```

```
===== END REQUEST =====
```

### A.4.10 Microsoft

Microsoft sends the key and the payload to two different Microsoft domains.

===== BEGIN REQUEST =====

```
PUT activity.windows.com/v2/feeds/me/activities/6fca2d91-9e74-3814-6fca-2d91
  ↪ 9e743814?
```

Content-Type: application/json

```
{..., "payload": "eyJiYWNRdXBUEXB1IjoiTVNBiwiY3JlYXRpb25EYXRlIjoiV2VkiEp1biA
  ↪ wMSAyMT01NDoyOSBQRfQgMjAyMiIsImRldmljZUlkZW50aWZpZXIiOiI4MDdhMTRlZiOzNGV
  ↪ mLTRlODgtYjAwZS04M2FmMGU1NzY3MWUiLCJkZXZpY2VOYW1lIjoiUGl4ZWwgM2EiLCJlbmN
  ↪ yeXB0ZWRCYWNRdXAiOiJleUpoYkdjaU9pSkJNVEk0UzFjaUxDSmxibU1pT2lKQk1USTRRMEp
  ↪ ETFVoVe1qVTJJbjAuT3BzdjkzM2pMOEhQakZtWTZ6UFREUdEtamZvUVd0MEI5VW5XOWxuZV9
  ↪ ZWFcza0ExOFBBbnRRLnpsU1JscERvdU4yT3R5eTlhc2JBclEuOFVXOGtrSVZrRjQ5X0p1NUV
  ↪ mQ28zdXNuQzFqaEZscTA1Vm03Uml0cE9tMjdXTERoXzlkZzczOXp0RXluRmJobExnaEpBbzN
  ↪ rTEJoVTU1UFg0TXlUcFNtZDhMN2t1MENWbGswSk4tbUp2dW1XU0RfRDfQVmkY3E3SDRcPcWI
  ↪ 2NHdzcXg4UGlCZDU5aXU4WGlzSFBOUkFsY2I2UGsxV1pmUWNwaHhacFhzcE4zU0x5ZERFMWd
  ↪ tR2xieEFuU9ucERzTk9xX0d5NWVKEu5DN2JIRXE2M25NajVvbXBvZEx2YjhFZjNlNFJyczB
  ↪ hbVpCaElVU1VxUmwtbW0yT1Y4SXl4ZVJ6bU8yZlVyYkEJ6M0UxbUtvS1l1QQWZqUE1WTjZ4UU5
  ↪ XZUdnU0FMZkhSTjNOWklaMOR3bFJ1eUMwU1ZrOEZFMkNIUUtqWWlnTGVDYWyWEJEU1JFSGh
  ↪ 5Nnp3YkZmakNvaFoyTld1cTNqaUpZQ1l1LN2FLZ3c1TVhwVXh4bJhT1c4d3Q3d3JNbXlaTzd
  ↪ CZXcyUEVOMW15X2N6TnBXNkZnQjJIRVJhMUh6MWFJWm9MMjhPdEwOYldKZ0puSzVKNU1NRlh
  ↪ KVHBYR050bDgyUElrbVJGUkYOMGFuakhPd2R5WkVEQUFsNnFUcktRRnhUZVZLNVE0bEp1Smk
  ↪ ycmZORUpnaElJY1hYRGnjUDQxNG1tYnV3TGw4NEZzaWNGZnMyYTnpCUp4SFJJRmFfS1ZiVXB
  ↪ NUVBGM3daMWVDTjdmX3dUZkd1UUI5TlRyb1RCc3lReUIwVnk14SHBDcTBSZlpNqnBoZWtYUnR
  ↪ KQkRXU2VQNfNDcWlIVnB3VnRXOEptWTRNZHRSb3djZFlhQm5wMUtucXpxWkJoSkxRSzZtd1B
  ↪ OWG1LYVZBQkFCV2hobjQtV2lUdGh3a1R5Nm0tYUVtZk9wLWNjekRpOU5GS1F3QWg5bEhCaWV
  ↪ wbFY4dmhTdkFPRnBYSVd0eFRFem5NNVRZegT5NkEzbTV3cWozM29oYmJyb0YyVvNjRi1tZEU
  ↪ tNERiVWFQRVpJTXlkWWJvdFpja3BGS0Z2VHVzUkxzZb0NnUkU2VFJEMy1FY0t1Y3NhVnlMQ1l
  ↪ ONnRka3V2bUFYae5mV0ZwX21DN0tvMzYtVv9la0ZxcHNDWktuaVlieVd0dkdDae5kb1AwS1F
  ↪ hbmVDUFbVrjBucW9yaEh2dElMNzVmTzFSWwo1TmFCR3B6V3hBSWlSRmNWMEpSY0xSZ0ZNbnI
  ↪ 1bTM5WmZnemN10TBRbHphckM5UUlueDhLV0FvV1dfcGxqdk1HT1hua2w2dUNSMlcyOWYyT2V
  ↪ 6VW9Vek9LMFpJLVdFQkxMLXAzQkJrbDdYeHdLRm9ZTFk3NWhqTzFDTzIwYml3VmdTSFZ4N3B
  ↪ ZdFVCNTlQYml2cXBQemNwRC1NNEpkM0w0WDhfS3A0YVVKVjNWRDd3azlZXXZWGMGxsaUg0amJ
  ↪ qM0hqWHBEY2hTaFhZSG9ScUpEZnl6aVFQdXJSYnlab0NfZWZJamlobj1wRVpiUnQ2aVhHcnR
  ↪ meWd5RGU1SXZpVEloV3NjdTZpS1lhbnhHSV9vckgtbDg4Z3c4Q0J3TWN2XzV4Y21kVWpWdkR
  ↪ lVDIwU0xCbG1iUktPYkotVnBwUmlkekhuai1QT1lkbWU3V000eHFBUV9KQk5QTGktblMyVTV
  ↪ obVNxcnN4d0xESWcwZkJtSlvzeXkuNEtuNgTIYjBOM3lsUWFIZlZrQmV4QSI5Im1zYUtleU1
  ↪ kIjoimjAyMiOwNiOwMlQwND01NDoxN1oiLCJ1cGRhdGVkRGF0ZSI6IldlZCBKdW4gMDEgMjE
  ↪ 6NTQ6MzIgUERUIDIwMjIifQ\u003d\u003d", ...}
```

===== END REQUEST =====

Encryption key gets sent to the server as “KeyMaterial”.

```
===== BEGIN REQUEST =====
POST login.live.com/ppsecure/GetUserKeyData.srf?
Content-Type: application/x-www-form-urlencoded
...
===== END REQUEST =====
===== BEGIN RESPONSE =====
HTTP/1.1 200 OK
Content-Type: text/xml

<?xml version="1.0" encoding="utf-8" ?>
<S:Envelope><S:Body><ps:GetKeyDataResponse
  ↳ Success="true"><ps:KeyPurposes><ps:KeyPurpose
  ↳ name="StrongCredentialKey"><ps:KeyData><ps:Property name="KeyMaterial">S
  ↳ bQv5hX0zBCLw70reWvBX3PvmzChD1QL6UnpGFfG/5w=</ps:property>...
===== END RESPONSE =====
```

# Appendix B

## Supplemental Materials for Chapter 5

### B.1 2FA descriptions used in screening survey

See Section 5.1.1 and Figure 5.1. Multiple choice options were *Yes*, *No*, and *I do not know*.

When logging into a **personal, non-work account** within the past 1 month, did you ever enter your username/password and...

- enter a code that the website sent you via a flying puppy dog?
- enter a code that the website sends you via text message (SMS)?
- enter a code that the website sends you via email?
- enter a code that the website sends you via a phone call?
- tap a button on your mobile device after receiving a notification from the website asking if you are logging in?
- approve the use of a pass**key** (not a pass**word**) that you previously created for your account?
- enter a periodically changing code generated by an app on your mobile device / computer?
- press a button on a physical security key (e.g. a yubikey) that is connected to your device via USB, bluetooth, or NFC?
- enter a one-time use code from a list of codes that you previously printed out or downloaded?

## **B.2 Contents of Main Survey**

The pages of this section contain the main survey formatted by Qualtrics when exporting to Microsoft Word with the survey logic included.

```

EmbeddedData
  compensation_dollars_per_minute = .25
  survey_duration_minutes = 13
  compensation_amount = $$e{e://Field/compensation_dollars_per_minute *
e://Field/survey_duration_minutes}
  survey_duration = ${e://Field/survey_duration_minutes} minutes
  env = prod
  PROLIFIC_PIDValue will be set from Panel or URL.
  STUDY_IDValue will be set from Panel or URL.
  SESSION_IDValue will be set from Panel or URL.
  redirect_url_default = https://app.prolific.com/submissions/complete?cc=2C0AC20F
  redirect_url_no_consent =
https://app.prolific.com/submissions/complete?cc=CSC7THK0
  redirect_url_failed_multiple_attention_checks =
https://app.prolific.com/submissions/complete?cc=CFY9LE5I
  redirect_url_used_totp_app_only_for_work =
https://app.prolific.com/submissions/complete?cc=C186M5WA
  redirect_url_failed_comprehension_quiz =
https://app.prolific.com/submissions/complete?cc=C21Z0H11
  redirect_url_selected_never_for_all_apps =
https://app.prolific.com/submissions/complete?cc=C7SGN1XS
  failed_attention_check = false
  is_current_platform_android = false
  cloud_instructions_clicked = false
  is_google_personal = false
  is_microsoft_personal = false
  is_authy_personal = false
  is_duo_personal = false
  current_personal_count = 0
  prev_personal_count = 0
  is_microsoft_current = false
  is_microsoft_prev = false
  is_google_current = false
  is_google_prev = false
  is_authy_current = false
  is_authy_prev = false
  is_duo_current = false
  is_duo_prev = false
  most_familiar_app = DEFAULT
  most_familiar_app_raw = DEFAULT
  most_familiar_app_developer = DEFAULT
  failed_attention_check_count = 0
  failed_attention_check_backups_knowledge = false
  failed_attention_check_cloud_attention = false

```

<b>Block: Participation Consent Form (Full) (2 Questions)</b>
<b>Branch: New Branch</b> If If Consent to Participate in Research Introduction and Purpose My name is Conor Gilsenan. I am a... I choose <strong>not</strong> to participate. Is Selected
EmbeddedData survey_status = declined_to_participate
<b>EndSurvey: Advanced</b>
EmbeddedData survey_status = agreed_to_participate
<b>Standard: Define TOTP apps - Comprehension Questions (4 Questions)</b>
EmbeddedData survey_status = finished_comprehension_1
<b>Branch: New Branch</b> If If How long does a TOTP code typically last before rotating to a new value? 30 seconds Is Selected And What term will be used to refer to the apps that generate periodically changing login codes? TOTP apps Is Selected
EmbeddedData comprehension_questions_take_1 = passed
<b>Branch: New Branch</b> If If How long does a TOTP code typically last before rotating to a new value? 30 seconds Is Not Selected Or What term will be used to refer to the apps that generate periodically changing login codes? TOTP apps Is Not Selected
EmbeddedData comprehension_questions_take_1 = failed comprehension_questions_take_1_duration = \${q://QID135/ChoiceGroup/SelectedChoices} comprehension_questions_take_1_term = \${q://QID137/ChoiceGroup/SelectedChoices}
<b>Standard: Define TOTP apps - Retake (2 Questions)</b> <b>Block: Define TOTP apps - Comprehension Questions (4 Questions)</b>
EmbeddedData



survey_status = finished_comprehension_2
<b>Branch: New Branch</b> If If How long does a TOTP code typically last before rotating to a new value? 30 seconds Is Not Selected Or What term will be used to refer to the apps that generate periodically changing login codes? TOTP apps Is Not Selected
EmbeddedData comprehension_questions_take_2 = failed comprehension_questions_take_2_duration = \${q://QID135/ChoiceGroup/SelectedChoices} comprehension_questions_take_2_term = \${q://QID137/ChoiceGroup/SelectedChoices} survey_status = failed_comprehension_2
<b>EndSurvey: Advanced</b>
EmbeddedData comprehension_questions_take_2 = passed
<b>Block: Which TOTP apps have they used? - Grid (2 Questions)</b>
EmbeddedData survey_status = finished_which_apps
<b>Branch: New Branch</b> If If Have you ever used any of the following apps to generate the TOTP code required to login to a per... - I have never used this app to generate TOTP codes, or I am unsure Is Equal to 4
<b>EndSurvey: Advanced</b>
<b>Branch: New Branch</b> If If Have you ever used any of the following apps to generate the TOTP code required to login to a per... - I <strong>currently use</strong> this app to generate TOTP codes Is Greater Than 0 Or Have you ever used any of the following apps to generate the TOTP code required to login to a per... - I <strong>previously used</strong> this app to generate TOTP codes, but no longer do Is Greater Than 0
<b>Block: Account type / platform (carry forward) (4 Questions)</b>
EmbeddedData survey_status = finished_each_app

Branch: New Branch

If

If Which types of accounts have you registered in the following TOTP apps?

<span class="app\_logo\_container">Microsoft Authenticator<br /> <br /> </span> - Only personal accounts Is Selected

Or Which types of accounts have you registered in the following TOTP apps?

<span class="app\_logo\_container">Microsoft Authenticator<br /> <br /> </span> - Both personal <strong>and</strong> work accounts Is Selected

EmbeddedData

is\_microsoft\_personal = true

Branch: New Branch

If

If Have you ever used any of the following apps to generate the TOTP code required to login to a per... <span class="app\_logo\_container">Microsoft Authenticator<br /> <br /> </span> - I <strong>currently use</strong> this app to generate TOTP codes Is Selected

EmbeddedData

current\_personal\_count = \${e://Field/current\_personal\_count + 1}  
is\_microsoft\_current = true

Branch: New Branch

If

If Have you ever used any of the following apps to generate the TOTP code required to login to a per... <span class="app\_logo\_container">Microsoft Authenticator<br /> <br /> </span> - I <strong>previously used</strong> this app to generate TOTP codes, but no longer do Is Selected

EmbeddedData

prev\_personal\_count = \${e://Field/prev\_personal\_count + 1}  
is\_microsoft\_prev = true

Branch: New Branch

If

If Which types of accounts have you registered in the following TOTP apps?

<span class="app\_logo\_container">Google Authenticator<br /> <br /> </span> - Only personal accounts Is Selected

Or Which types of accounts have you registered in the following TOTP apps?  
 <span class="app\_logo\_container">Google Authenticator<br /> <br /> </span> - Both personal <strong>and</strong> work accounts Is Selected

EmbeddedData  
 is\_google\_personal = true

Branch: New Branch

If

If Have you ever used any of the following apps to generate the TOTP code required to login to a per... <span class="app\_logo\_container">Google Authenticator<br /> <br /> </span> - I <strong>currently use</strong> this app to generate TOTP codes Is Selected

EmbeddedData  
 current\_personal\_count = \${e://Field/current\_personal\_count + 1}  
 is\_google\_current = true

Branch: New Branch

If

If Have you ever used any of the following apps to generate the TOTP code required to login to a per... <span class="app\_logo\_container">Google Authenticator<br /> <br /> </span> - I <strong>previously used</strong> this app to generate TOTP codes, but no longer do Is Selected

EmbeddedData  
 prev\_personal\_count = \${e://Field/prev\_personal\_count + 1}  
 is\_google\_prev = true

Branch: New Branch

If

If Which types of accounts have you registered in the following TOTP apps?  
 <span class="app\_logo\_container">Authy<br /> <br /> </span> - Only personal accounts Is Selected

Or Which types of accounts have you registered in the following TOTP apps?  
 <span class="app\_logo\_container">Authy<br /> <br /> </span> - Both personal  
 <strong>and</strong> work accounts Is Selected

EmbeddedData  
 is\_authy\_personal = true

Branch: New Branch

If

If Have you ever used any of the following apps to generate the TOTP  
 code required to login to a per... <span class="app\_logo\_container">Authy<br />  
 <br /> </span> - I <strong>currently  
 use</strong> this app to generate TOTP codes Is Selected

EmbeddedData  
 current\_personal\_count = \${e://Field/current\_personal\_count + 1}  
 is\_authy\_current = true

Branch: New Branch

If

If Have you ever used any of the following apps to generate the TOTP  
 code required to login to a per... <span class="app\_logo\_container">Authy<br />  
 <br /> </span> - I <strong>previously  
 used</strong> this app to generate TOTP codes, but no longer do Is Selected

EmbeddedData  
 prev\_personal\_count = \${e://Field/prev\_personal\_count + 1}  
 is\_authy\_prev = true

Branch: New Branch

If

If Which types of accounts have you registered in the following TOTP apps?  
 <span class="app\_logo\_container">Duo Mobile<br /> <br /> </span> - Only personal accounts Is  
 Selected

Or Which types of accounts have you registered in the following TOTP apps?  
 <span class="app\_logo\_container">Duo Mobile<br /> <br /> </span> - Both personal<br><strong>and</strong> work accounts Is Selected   |
| EmbeddedData<br>is_duo_personal = true   |
| Branch: New Branch<br>If<br>If Have you ever used any of the following apps to generate the TOTP<br>code required to login to a per... <span class="app_logo_container">Duo<br>Mobile<br /> <br /> <img<br>src="https://berkeley.qualtrics.com/CP/Graphic.php?IM=IM_29oMwU2Fa3OXovb"<br>style="width:100%; max-width:250px;" /></span> - I <strong>currently<br>use</strong> this app to generate TOTP codes Is Selected                     |
| EmbeddedData<br>current_personal_count = \${e://Field/current_personal_count + 1}<br>is_duo_current = true   |
| Branch: New Branch<br>If<br>If Have you ever used any of the following apps to generate the TOTP<br>code required to login to a per... <span class="app_logo_container">Duo<br>Mobile<br /> <br /> <img<br>src="https://berkeley.qualtrics.com/CP/Graphic.php?IM=IM_29oMwU2Fa3OXovb"<br>style="width:100%; max-width:250px;" /></span> - I <strong>previously<br>used</strong> this app to generate TOTP codes, but no longer do Is Selected |
| EmbeddedData<br>prev_personal_count = \${e://Field/prev_personal_count + 1}<br>is_duo_prev = true  |
| Branch: New Branch<br>If<br>If prev_personal_count Is Equal to 0<br>And current_personal_count Is Equal to 0   |
| EndSurvey: Advanced  |
| Branch: New Branch<br>If<br>If prev_personal_count Is Greater Than 0   |
| Block: Why did they stop using some apps? (7 Questions)  |
| EmbeddedData<br>survey_status = finished_previous_apps   |

<b>Standard: Lockout knowledge (3 Questions)</b>
<b>EmbeddedData</b> survey_status = finished_lockout_knowledge
<b>Branch: New Branch</b> If If current_personal_count Is Greater Than 0
<b>Standard: Pick most familiar app (3 Questions)</b>
<b>EmbeddedData</b> most_familiar_app_raw = \${q://QID228/ChoiceGroup/SelectedChoices}
<b>Branch: New Branch</b> If If most_familiar_app_raw Contains Microsoft
<b>EmbeddedData</b> most_familiar_app = Microsoft Authenticator most_familiar_app_developer = Microsoft
<b>Branch: New Branch</b> If If On which type of device(s) have you used each of the following TOTP apps? Select all that apply. <span class="app_logo_container">Microsoft Authenticator    </span> - Android phone/tablet Is Selected
<b>EmbeddedData</b> is_current_platform_android = true who_can_read = the developer (Microsoft)
<b>Branch: New Branch</b> If If most_familiar_app_raw Contains Google
<b>EmbeddedData</b> most_familiar_app = Google Authenticator most_familiar_app_developer = Google
<b>Branch: New Branch</b> If If On which type of device(s) have you used each of the following TOTP apps? Select all that apply. <span class="app_logo_container">Google Authenticator    </span> - Android phone/tablet Is
Selected
```

```
EmbeddedData
  is_current_platform_android = true
  who_can_read = the developer (Google)
```

Branch: New Branch

```
If
  If most_familiar_app_raw Contains Duo
```

```
EmbeddedData
  most_familiar_app = Duo Mobile
  most_familiar_app_developer = Duo Security
```

Branch: New Branch

```
If
  If On which type of device(s) have you used each of the following TOTP
  apps? Select all that apply. <span class="app_logo_container">Duo Mobile<br />
  <br /> </span> - Android phone/tablet Is
  Selected
```

```
EmbeddedData
  is_current_platform_android = true
  who_can_read = Google (via Google Drive)
```

Branch: New Branch

```
If
  If most_familiar_app_raw Contains Authy
```

```
EmbeddedData
  most_familiar_app = Authy
  most_familiar_app_developer = Twilio
```

Branch: New Branch

```
If
  If On which type of device(s) have you used each of the following TOTP
  apps? Select all that apply. <span class="app_logo_container">Authy<br /> <br
  /> </span> - Android phone/tablet Is
  Selected
```

```
EmbeddedData
  is_current_platform_android = true
```





|                                                                                                                                                                                                          |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>EmbeddedData</b><br>survey_status = finished_cloud_guess_and_check                                                                                                                                    |
| <b>Branch: New Branch</b><br>If<br>If The previous question asked you to follow the instructions in a linked Google Doc. What is the at... Text Response Does Not Contain purple                         |
| <b>EmbeddedData</b><br>failed_attention_check = true<br>failed_attention_check_cloud_attention = true<br>failed_attention_check_count = \${e://Field/failed_attention_check_count + 1}                   |
| <b>Branch: New Branch</b><br>If<br>If failed_attention_check_count Is Greater Than 1                                                                                                                     |
| <b>EndSurvey: Advanced</b>                                                                                                                                                                               |
| <b>Branch: New Branch</b><br>If<br>If Please follow the instructions in this Google Doc to determine whether the cloud-based backup fea... Cloud-based backups are <strong>enabled</strong> Is Selected  |
| Standard: Cloud backups are ENABLED (6 Questions)                                                                                                                                                        |
| <b>EmbeddedData</b><br>survey_status = finished_cloud_enabled                                                                                                                                            |
| <b>Branch: New Branch</b><br>If<br>If Please follow the instructions in this Google Doc to determine whether the cloud-based backup fea... Cloud-based backups are <strong>disabled</strong> Is Selected |
| Standard: Cloud backups are DISABLED (5 Questions)                                                                                                                                                       |
| <b>EmbeddedData</b><br>survey_status = finished_cloud_disabled                                                                                                                                           |
| Standard: Precautionary TOTP backup actions (recovery codes and setup QR code) (12 Questions)                                                                                                            |
| <b>EmbeddedData</b><br>survey_status = finished_other_backups<br>likely_to_continue_using = \${q://QID104/ChoiceGroup/SelectedAnswers}                                                                   |
| Block: Leaky Backups - Current Users (7 Questions)                                                                                                                                                       |

|                                                                                                                                                           |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| EmbeddedData<br>survey_status = finished_leaky_backups                                                                                                    |
| Standard: Account Lockout / Recovery - Part 1 (3 Questions)                                                                                               |
| EmbeddedData<br>survey_status = finished_lockout_1                                                                                                        |
| Branch: New Branch<br>If<br>If How often have you been unable to provide a TOTP code within 5 minutes of being prompted to enter... Never Is Not Selected |
| Standard: Account Lockout / Recovery - Part 2 (7 Questions)                                                                                               |
| EmbeddedData<br>survey_status = finished_lockout_2                                                                                                        |
| Standard: Demographic Questions (8 Questions)                                                                                                             |
| EmbeddedData<br>survey_status = completed                                                                                                                 |
| Standard: Thanks, etc (2 Questions)                                                                                                                       |
| EndSurvey: Advanced                                                                                                                                       |

Page Break

---

Start of Block: Participation Consent Form (Full)

consent **Consent to Participate in Research** Introduction and Purpose My name is Conor Gilsenan. I am a graduate student at the University of California, Berkeley working with my faculty advisor, Serge Egelman, in the Electrical Engineering and Computer Science (EECS) department and affiliated with the International Computer Science Institute (ICSI). I would like to invite you to take part in our research study, which aims to understand people's experiences protecting their online accounts. Procedures The survey should take about `#{e://Field/survey_duration}` to complete. Benefits There is no direct benefit to you from taking part in this study. Risks/Discomforts There are no known risks of participating in this research and you are free to stop participating at any time. Confidentiality This survey does not collect any personally identifiable information (PII). Your Prolific ID is recorded so that we can compensate you for participating. When the research is completed, your responses will be saved indefinitely and made publicly available for possible use in future research done by the authors of this study, or others. Compensation Participants will receive `#{e://Field/compensation_amount}` credited to their Prolific account when their submission is approved. Submissions by participants who have clearly made little effort, failed multiple attention checks, or have taken other malicious/fraudulent actions will be rejected and will not receive compensation. Rights Participation in research is completely voluntary. You are free to decline to take part in the project. If you choose to participate, you are free to stop taking part in the project at any time. Whether or not you choose to participate, to answer any particular question, or continue participating in the project, there will be no penalty to you or loss of benefits to which you are otherwise entitled. Questions If you have any questions about this research, please feel free to contact Conor Gilsenan ([conorgilsenan@berkeley.edu](mailto:conorgilsenan@berkeley.edu)) and/or Serge Egelman ([egelman@berkeley.edu](mailto:egelman@berkeley.edu)).

- ☐ I have read the above Consent form and **I agree to participate** in this research. (1)
- ☐ I choose **not** to participate. (2)
- 

consent\_t Timing

First Click (1)

Last Click (2)

Page Submit (3)

Click Count (4)

---

End of Block: Participation Consent Form (Full)

---

Start of Block: Define TOTP apps - Comprehension Questions

intro **Introduction**

If you do not meet the following criteria, please return your submission in Prolific.

In a previous survey, you indicated that you have entered your username/password and a periodically changing code generated by an app on your mobile device / computer when logging into a **personal, non-work account**. These periodically changing login codes are called time-based one-time passwords (TOTPs) and typically rotate to a new value every 30 seconds. This survey will refer to the apps that generate TOTP codes as "TOTP apps." Here is an example of what a TOTP app might look like:



quiz\_30seconds How long does a TOTP code typically last before rotating to a new value?

- ☐ 1 second (2)
- ☐ 30 seconds (1)
- ☐ 1 minute (7)
- ☐ 5 minutes (5)
- ☐ 30 minutes (3)



quiz\_term What term will be used to refer to the apps that generate periodically changing login codes?

- ☐ TOTP apps (1)
  - ☐ Authenticators (2)
  - ☐ Login apps (3)
  - ☐ 2FA apps (4)
-

quiz\_t Timing  
First Click (1)  
Last Click (2)  
Page Submit (3)  
Click Count (4)

End of Block: Define TOTP apps - Comprehension Questions

---

Start of Block: Define TOTP apps - Retake



quiz\_failed Unfortunately, you did not answer all of the comprehension questions correctly.  
Please read the Introduction more carefully and answer these questions again.

---

quiz\_failed\_t Timing  
First Click (1)  
Last Click (2)  
Page Submit (3)  
Click Count (4)

End of Block: Define TOTP apps - Retake

---

Start of Block: Which TOTP apps have they used? - Grid



app\_used Have you ever used any of the following apps to generate the TOTP code required to login to a **personal, non-work account**?

|                                                 | I <b>currently use</b> this app to generate TOTP codes (4) | I <b>previously used</b> this app to generate TOTP codes, but no longer do (5) | I have never used this app to generate TOTP codes, or I am unsure (6) |
|-------------------------------------------------|------------------------------------------------------------|--------------------------------------------------------------------------------|-----------------------------------------------------------------------|
| Microsoft Authenticator<br>(app_used_microsoft) | <input type="radio"/>                                      | <input type="radio"/>                                                          | <input type="radio"/>                                                 |
| Google Authenticator<br>(app_used_google)       | <input type="radio"/>                                      | <input type="radio"/>                                                          | <input type="radio"/>                                                 |
| Duo Mobile<br>(app_used_duo)                    | <input type="radio"/>                                      | <input type="radio"/>                                                          | <input type="radio"/>                                                 |
| Authy<br>(app_used_authy)                       | <input type="radio"/>                                      | <input type="radio"/>                                                          | <input type="radio"/>                                                 |

app\_used\_t Timing

First Click (1)

Last Click (2)

Page Submit (3)

Click Count (4)

End of Block: Which TOTP apps have they used? - Grid

Start of Block: Account type / platform (carry forward)

Carry Forward Unselected Choices from "app\_used"



account\_type Which types of accounts have you registered in the following TOTP apps?

|                                                     | Only personal<br>accounts (1) | Only work<br>accounts (2) | Both personal<br><b>and</b> work<br>accounts (3) | I do not know<br>(4)  |
|-----------------------------------------------------|-------------------------------|---------------------------|--------------------------------------------------|-----------------------|
| Microsoft Authenticator<br>(account_type_microsoft) | <input type="radio"/>         | <input type="radio"/>     | <input type="radio"/>                            | <input type="radio"/> |
| Google Authenticator<br>(account_type_google)       | <input type="radio"/>         | <input type="radio"/>     | <input type="radio"/>                            | <input type="radio"/> |
| Duo Mobile<br>(account_type_duo)                    | <input type="radio"/>         | <input type="radio"/>     | <input type="radio"/>                            | <input type="radio"/> |
| Authy<br>(account_type_authy)                       | <input type="radio"/>         | <input type="radio"/>     | <input type="radio"/>                            | <input type="radio"/> |

account\_type\_t Timing

First Click (1)

Last Click (2)

Page Submit (3)

Click Count (4)

Page Break

Carry Forward Unselected Choices from "app\_used"



platform On which type of device(s) have you used each of the following TOTP apps? Select all that apply.

|                                                    | Android<br>phone/tablet (1) | Apple<br>phone/tablet (2) | Other (3)                | I do not know<br>(4)     |
|----------------------------------------------------|-----------------------------|---------------------------|--------------------------|--------------------------|
| Microsoft<br>Authenticator<br>(platform_microsoft) | <input type="checkbox"/>    | <input type="checkbox"/>  | <input type="checkbox"/> | <input type="checkbox"/> |
| Google<br>Authenticator<br>(platform_google)       | <input type="checkbox"/>    | <input type="checkbox"/>  | <input type="checkbox"/> | <input type="checkbox"/> |
| Duo Mobile<br>(platform_duo)                       | <input type="checkbox"/>    | <input type="checkbox"/>  | <input type="checkbox"/> | <input type="checkbox"/> |
| Authy<br>(platform_authy)                          | <input type="checkbox"/>    | <input type="checkbox"/>  | <input type="checkbox"/> | <input type="checkbox"/> |

platform\_t Timing

First Click (1)

Last Click (2)

Page Submit (3)

Click Count (4)

End of Block: Account type / platform (carry forward)

Start of Block: Why did they stop using some apps?



Display this question:

*If is\_microsoft\_personal = true*

*And is\_microsoft\_prev = true*

Or If

*is\_google\_personal = true*

*And is\_google\_prev = true*

Or If

*is\_duo\_personal = true*

*And is\_duo\_prev = true*

Or If

*is\_authy\_personal = true*

*And is\_authy\_prev = true*

#### Q387 Previously Used Apps

Display this question:

*If is\_microsoft\_personal = true*

*And is\_microsoft\_prev = true*

#### Q383 Microsoft Authenticator

Display this question:

*If is\_google\_personal = true*

*And is\_google\_prev = true*

#### Q384 Google Authenticator

Display this question:

*If is\_duo\_personal = true*

*And is\_duo\_prev = true*

#### Q385 Duo Mobile

Display this question:

If *is\_authy\_personal* = true  
And *is\_authy\_prev* = true

Q386 Authy

Display this question:

If *is\_microsoft\_personal* = true  
And *is\_microsoft\_prev* = true  
Or If  
    *is\_google\_personal* = true  
    And *is\_google\_prev* = true  
Or If  
    *is\_duo\_personal* = true  
    And *is\_duo\_prev* = true  
Or If  
    *is\_authy\_personal* = true  
    And *is\_authy\_prev* = true

**why\_stopped\_using** Why did you stop using the listed apps to generate TOTP codes for your **personal, non-work account(s)**?

---

---

---

---

---

**why\_stopped\_using\_t** Timing

First Click (1)

Last Click (2)

Page Submit (3)

Click Count (4)

End of Block: Why did they stop using some apps?

---

**Start of Block: Lockout knowledge****Q8.1 General Usage Questions**

lockout\_knowledge Read the following scenario and answer the question below: Alice can login to her account with just a username/password. The site has no customer support, but does allow users to reset their password using a 'Forgot password' link. Alice takes the following steps to enable TOTP on her account: 1) Installs a TOTP app that can only add new accounts and display TOTP codes 2) Scans the setup QR code displayed in the browser with the TOTP app 3) Types a TOTP code into the browser to complete the setup process **If Alice loses access to her TOTP app, will she be able to generate the TOTP codes required to login to her account?**

- ☐ Yes (1)
- ☐ No (2)
- ☐ I do not know (3)

---

lockout\_knowledge\_t Timing

First Click (1)

Last Click (2)

Page Submit (3)

Click Count (4)

---

**End of Block: Lockout knowledge**

---

**Start of Block: Pick most familiar app****Q359 Currently Used Apps**

q\_most\_familiar\_app Which of these TOTP app(s) that you currently use for **personal, non-work accounts** are you the most familiar with?

Display this choice:

If is\_microsoft\_personal = true

And is\_microsoft\_current = true

☐ Microsoft Authenticator (33)

Display this choice:

If is\_google\_personal = true

And is\_google\_current = true

☐ Google Authenticator (36)

Display this choice:

If is\_duo\_personal = true

And is\_duo\_current = true

☐ Duo Mobile (37)

Display this choice:

If is\_authy\_personal = true

And is\_authy\_current = true

☐ Authy (38)

most\_familiar\_app\_t Timing

First Click (1)

Last Click (2)

Page Submit (3)

Click Count (4)

End of Block: Pick most familiar app

Start of Block: Questions about most familiar app they CURRENTLY use - Part 1

Q11.1 Questions about \${e://Field/most\_familiar\_app}



trust\_before Please rate how you feel about the following statement. "I trust `#{e://Field/most_familiar_app_developer}`, the developer of the `#{e://Field/most_familiar_app}` app."

|                  | Strongly<br>Disagree (1) | Disagree (2)          | Neutral (3)           | Agree (4)             | Strongly<br>Agree (5) |
|------------------|--------------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| (trust_before_1) | <input type="radio"/>    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

trust\_before\_t Timing  
First Click (1)  
Last Click (2)  
Page Submit (3)  
Click Count (4)



login\_confidence If you lost *all* of your devices on which the app is installed, how confident are you that you would be able to login to all of your **personal, non-work accounts** that are registered in [\\${e://Field/most\\_familiar\\_app}](#)?

|     | I definitely<br>would not be<br>able to login<br>to all<br>accounts (1) | I likely<br>would not be<br>able to login<br>to all<br>accounts (2) | Unsure (3)            | I likely<br>would be<br>able to login<br>to all<br>accounts (4) | I definitely<br>would be<br>able to login<br>to all<br>accounts (5) |
|-----|-------------------------------------------------------------------------|---------------------------------------------------------------------|-----------------------|-----------------------------------------------------------------|---------------------------------------------------------------------|
| (6) | <input type="radio"/>                                                   | <input type="radio"/>                                               | <input type="radio"/> | <input type="radio"/>                                           | <input type="radio"/>                                               |

login\_confidence\_t Timing  
First Click (1)  
Last Click (2)  
Page Submit (3)  
Click Count (4)

login\_how How would you attempt to login to those **personal, non-work** accounts?

---

---

---

---

---

-----

login\_how\_t Timing  
First Click (1)  
Last Click (2)  
Page Submit (3)  
Click Count (4)

-----

Page Break 

---



num\_devices How many devices do you currently have [\\${e://Field/most\\_familiar\\_app}](#) installed on?

- ☐ 1 (1)
- ☐ 2 (2)
- ☐ 3 (3)
- ☐ 4 (4)
- ☐ 5 or more (98)
- ☐ I do not know (-99)

---

num\_devices\_t Timing  
First Click (1)  
Last Click (2)  
Page Submit (3)  
Click Count (4)

---

Page Break





backups\_knowledge Some TOTP apps contain backup features that may help you maintain your ability to generate the TOTP codes required to login to your account(s) if you lose your device(s). To the best of your knowledge, **without looking at the app**, does

[\\${e://Field/most\\_familiar\\_app}](#) support the following features?

|                                                                                                                                                                  | Yes (1)               | No (2)                | I do not know (3)     |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|-----------------------|-----------------------|
| <b>Cloud-based Backups</b> The app can backup your accounts to a remote storage service.<br>(backups_knowledge_cloud)                                            | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| <b>Transfer via QR Code</b> The app can display a QR code that allows you to manually transfer all of your accounts to another device.<br>(backups_knowledge_qr) | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| <b>Attention Check</b> Select 'Yes' as your answer.<br>(backups_knowledge_attention_check)                                                                       | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

backups\_knowledge\_t Timing

First Click (1)

Last Click (2)

Page Submit (3)

Click Count (4)

End of Block: Questions about most familiar app they CURRENTLY use - Part 1

Start of Block: QR backups



qr\_backup\_used [\\${e://Field/most\\_familiar\\_app}](#) does allow users to manually transfer the accounts registered in the app to another device using QR codes. Have you ever used this feature?

- ☐ Yes (1)
- ☐ No (2)
- ☐ I do not know (3)

---

qr\_backup\_used\_t Timing  
First Click (1)  
Last Click (2)  
Page Submit (3)  
Click Count (4)

---

Page Break

Display this question:

If `qr_backup_used = 1`



qr\_all\_backed\_up How much do you agree or disagree with the following statement? "All of my accounts that are registered in `{e://Field/most_familiar_app}` have been manually transferred to another device using QR codes"

|     | Strongly disagree (1) | Disagree (2)          | Neither agree nor disagree (3) | Agree (4)             | Strongly Agree (5)    |
|-----|-----------------------|-----------------------|--------------------------------|-----------------------|-----------------------|
| (2) | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>          | <input type="radio"/> | <input type="radio"/> |

qr\_all\_backed\_up\_t Timing

First Click (1)

Last Click (2)

Page Submit (3)

Click Count (4)

End of Block: QR backups

Start of Block: Cloud guess and check in the app



cloud\_guess `{e://Field/most_familiar_app}` does support a cloud-based backup feature.

**Without looking at the app**, do you believe that cloud-based backups are enabled or disabled in your `{e://Field/most_familiar_app}` app?

|     | Cloud backups are definitely disabled (0) | Cloud backups are likely disabled (0) | Unsure (1)            | Cloud backups are likely enabled (2) | Cloud backups are definitely enabled (2) |
|-----|-------------------------------------------|---------------------------------------|-----------------------|--------------------------------------|------------------------------------------|
| (1) | <input type="radio"/>                     | <input type="radio"/>                 | <input type="radio"/> | <input type="radio"/>                | <input type="radio"/>                    |

cloud\_guess\_t Timing  
First Click (1)  
Last Click (2)  
Page Submit (3)  
Click Count (4)

-----  
Page Break



cloud\_is\_enabled Please follow the instructions in this Google Doc to determine whether the cloud-based backup feature is currently enabled or disabled in your [\\${e://Field/most\\_familiar\\_app}](#) app. **The survey question on the next page is an attention check that will ask you to enter the attention word listed in the linked instructions.** There is no back button, so please be sure to open and follow the linked instructions. Without changing the setting, is the cloud-based backup feature currently enabled or disabled in [\\${e://Field/most\\_familiar\\_app}](#)?

- ☐ Cloud-based backups are **enabled** (3)
- ☐ Cloud-based backups are **disabled** (4)
- ☐ I was not able to check whether cloud backups are enabled or disabled because... (5)
- 

---

cloud\_is\_enabled\_t Timing  
First Click (1)  
Last Click (2)  
Page Submit (3)  
Click Count (4)

---

Page Break

---

cloud\_attention The previous question asked you to follow the instructions in a linked Google Doc. What is the attention word documented in that Google Doc?

---

cloud\_attention\_t Timing  
First Click (1)  
Last Click (2)  
Page Submit (3)  
Click Count (4)

---

Page Break



accounts\_importance How important do you consider the **personal, non-work accounts** that you have registered in [\\${e://Field/most\\_familiar\\_app}](#)?

|     | Very<br>unimportant<br>(1) | Unimportant<br>(2)    | Neither<br>unimportant<br>nor important<br>(3) | Important (4)         | Very<br>important (5) |
|-----|----------------------------|-----------------------|------------------------------------------------|-----------------------|-----------------------|
| (5) | <input type="radio"/>      | <input type="radio"/> | <input type="radio"/>                          | <input type="radio"/> | <input type="radio"/> |

accounts\_t Timing

First Click (1)

Last Click (2)

Page Submit (3)

Click Count (4)

End of Block: Cloud guess and check in the app

Start of Block: Cloud backups are ENABLED



cloud\_enabled\_why Why did you choose to enable cloud-based backups?

---



---



---



---



---

cloud\_enabled\_why\_t Timing  
First Click (1)  
Last Click (2)  
Page Submit (3)  
Click Count (4)

---

Page Break





cloud\_ever\_used Have you ever used the cloud-based backup mechanism to recover your TOTP codes?

- ☐ Yes (1)
- ☐ No (2)
- ☐ I do not know (3)



cloud\_enabled\_keep Do you plan to keep cloud-based backups enabled?

|     | Definitely will<br><b>disable</b> (1) | Likely will<br><b>disable</b> (2) | Unsure (3)            | Likely will<br>leave<br><b>enabled</b> (4) | Definitely will<br>leave<br><b>enabled</b> (5) |
|-----|---------------------------------------|-----------------------------------|-----------------------|--------------------------------------------|------------------------------------------------|
| (1) | <input type="radio"/>                 | <input type="radio"/>             | <input type="radio"/> | <input type="radio"/>                      | <input type="radio"/>                          |

Display this question:

If most\_familiar\_app Contains Authy



cloud\_know\_pwd \${e://Field/most\_familiar\_app} requires choosing a backup password when cloud-based backups are enabled. How confident are you that you know the backup password?

|                    | I <b>definitely</b><br><b>do not</b><br><b>know</b> the<br>backup<br>password<br>(1) | I <b>likely do</b><br><b>not</b><br><b>know</b> the<br>backup<br>password<br>(2) | I am<br>unsure (3)    | I <b>likely</b><br><b>know</b> the<br>backup<br>password<br>(4) | I <b>definitely</b><br><b>know</b> the<br>backup<br>password<br>(5) |
|--------------------|--------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|-----------------------|-----------------------------------------------------------------|---------------------------------------------------------------------|
| (cloud_know_pwd_1) | <input type="radio"/>                                                                | <input type="radio"/>                                                            | <input type="radio"/> | <input type="radio"/>                                           | <input type="radio"/>                                               |

cloud\_enabled\_misc\_t Timing

First Click (1)

Last Click (2)

Page Submit (3)

Click Count (4)

End of Block: Cloud backups are ENABLED

---

Start of Block: Cloud backups are DISABLED

cloud\_disabled\_why Why did you choose not to enable (or to disable) cloud-based backups?

---

---

---

---

---

---

cloud\_disabled\_why\_t Timing

First Click (1)

Last Click (2)

Page Submit (3)

Click Count (4)

---

Page Break

---



cloud\_ever\_enabled Have you ever enabled the cloud-based backup feature in the past?

- ☐ Yes (1)
- ☐ No (2)
- ☐ I do not know (3)



cloud\_will\_enable Do you think you will enable the cloud-based backup feature in the future?

|     | I very likely<br>will not<br>enable it (1) | I likely will<br>not enable it<br>(2) | Undecided<br>(3)      | I likely will<br>enable it (4) | I very likely<br>will enable it<br>(5) |
|-----|--------------------------------------------|---------------------------------------|-----------------------|--------------------------------|----------------------------------------|
| (2) | <input type="radio"/>                      | <input type="radio"/>                 | <input type="radio"/> | <input type="radio"/>          | <input type="radio"/>                  |

cloud\_disabled\_t Timing

First Click (1)

Last Click (2)

Page Submit (3)

Click Count (4)

End of Block: Cloud backups are DISABLED

Start of Block: Precautionary TOTP backup actions (recovery codes and setup QR code)



recovery\_code\_has When first enabling TOTP on an account, some websites will display account recovery codes for you to save in case you lose access to your TOTP app. For example: Did you save any such account recovery codes when enabling TOTP on your **personal, non-work account(s)**?

- ☐ Yes, I saved account recovery codes for all of my personal, non-work accounts (1)
  - ☐ Yes, I saved account recovery codes for some of my personal, non-work accounts (4)
  - ☐ No (2)
  - ☐ I do not know (3)
- 

recovery\_code\_has\_t Timing

First Click (1)

Last Click (2)

Page Submit (3)

Click Count (4)

---

Page Break


Display this question:

If recovery\_code\_has = 1

Or recovery\_code\_has = 4



rc\_where\_stored Where did you store those account recovery code(s)? Select all that apply.

- ☐ I manually wrote them on a piece of paper (1)
- ☐ I printed them onto a piece of paper (2)
- ☐ I saved them to my computer (3)
- ☐ I saved them to my phone / mobile device (4)
- ☐ I saved them to my password manager (5)
- ☐ I saved them to my cloud storage (e.g., Google Drive, iCloud, etc) (6)
- ☒  do not know (7)
- ☐ Other... (8) \_\_\_\_\_

Display this question:

If recovery\_code\_has = 1

Or recovery\_code\_has = 4



rc\_can\_access How confident are you that you would be able to access those account recovery code(s) if you lost all of the devices on which [\\${e://Field/most\\_familiar\\_app}](#) is installed?

|     | I definitely<br>could not<br>access the<br>recovery<br>codes (1) | I likely could<br>not access<br>the recovery<br>codes (2) | I am unsure<br>(3)    | I likely could<br>access the<br>recovery<br>codes (4) | I definitely<br>could access<br>the recovery<br>codes (5) |
|-----|------------------------------------------------------------------|-----------------------------------------------------------|-----------------------|-------------------------------------------------------|-----------------------------------------------------------|
| (6) | <input type="radio"/>                                            | <input type="radio"/>                                     | <input type="radio"/> | <input type="radio"/>                                 | <input type="radio"/>                                     |

---

recovery\_code\_yes\_t Timing

First Click (1)

Last Click (2)

Page Submit (3)

Click Count (4)

---

Page Break



setup\_qr\_has When first enabling TOTP on your account, you are typically required to use your TOTP app to scan a setup QR code displayed by the website. For example: Did you save a screenshot of the setup QR code for any of your **personal, non-work accounts**?

- ☐ Yes, I saved the setup QR code for all of my personal, non-work accounts (1)
  - ☐ Yes, I saved the setup QR code for some of my personal, non-work accounts (4)
  - ☐ No (2)
  - ☐ I do not know (3)
- 

setup\_qr\_has\_t Timing

First Click (1)

Last Click (2)

Page Submit (3)

Click Count (4)

---

Page Break

---

Display this question:

If setup\_qr\_has = 1

Or setup\_qr\_has = 4



setup\_qr\_where Where did you save the setup QR code(s)? Select all that apply.

- ☐ I printed the setup QR code(s) on paper (1)
- ☐ I saved a screenshot of the setup QR code(s) on my phone / mobile device (2)
- ☐ I saved a screenshot of the setup QR code(s) on my laptop / computer (3)
- ☐ I saved the setup QR code(s) in my password manager (4)
- ☐ I saved the setup QR code(s) to my cloud storage (e.g., Google Drive, iCloud, etc) (5)
- ☐ Other... (6) \_\_\_\_\_
- ☒ do not know (7)

Display this question:

If setup\_qr\_has = 1

Or setup\_qr\_has = 4



setup\_qr\_access How confident are you that you would be able to access the setup QR code(s) if you lost all of the devices on which [\\${e://Field/most\\_familiar\\_app}](#) is installed?

|     | I definitely<br>could not<br>access the<br>setup QR<br>code (1) | I likely could<br>not access<br>the setup QR<br>code (2) | I am unsure<br>(3)    | I likely could<br>access the<br>setup QR<br>code (4) | I definitely<br>could access<br>the setup QR<br>code (5) |
|-----|-----------------------------------------------------------------|----------------------------------------------------------|-----------------------|------------------------------------------------------|----------------------------------------------------------|
| (6) | <input type="radio"/>                                           | <input type="radio"/>                                    | <input type="radio"/> | <input type="radio"/>                                | <input type="radio"/>                                    |



setup\_qr\_yes\_t Timing  
First Click (1)  
Last Click (2)  
Page Submit (3)  
Click Count (4)

Page Break

---



likely\_use\_before How likely are you to continue using [\\${e://Field/most\\_familiar\\_app}](#) to generate the TOTP codes required to login to your personal, non-work account(s)?

|     | Very unlikely<br>(1)  | Unlikely (2)          | Undecided<br>(3)      | Likely (4)            | Very Likely<br>(5)    |
|-----|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| (2) | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

likely\_use\_before\_t Timing

First Click (1)

Last Click (2)

Page Submit (3)

Click Count (4)

End of Block: Precautionary TOTP backup actions (recovery codes and setup QR code)

Start of Block: Leaky Backups - Current Users

#### Q12.1 Cloud Backup Questions



all\_expectation\_dev Do you believe the developer of [\\${e://Field/most\\_familiar\\_app}](#) can read the following data from the cloud-based backups created by the app?

|                                                                                                        | Yes (1)               | No (2)                | I do not know (3)     |
|--------------------------------------------------------------------------------------------------------|-----------------------|-----------------------|-----------------------|
| The <b>name of the website</b> on which each account was created (all_expectation_dev_website)         | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| The <b>username</b> for each account (all_expectation_dev_username)                                    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| The <b>secrets</b> required to generate valid TOTP codes for each account (all_expectation_dev_secret) | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |



all\_expectation\_3rd Do you believe other third parties (*not* the developer of `${e://Field/most_familiar_app}`) can read the following data from the cloud-based backups created by the app?

|                                                                                                           | Yes (1)               | No (2)                | I do not know (3)     |
|-----------------------------------------------------------------------------------------------------------|-----------------------|-----------------------|-----------------------|
| The <b>name of the website</b> on which each account was created<br>(all_expectation_3rd_website)         | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| The <b>username</b> for each account<br>(all_expectation_3rd_username)                                    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| The <b>secrets</b> required to generate valid TOTP codes for each account<br>(all_expectation_3rd_secret) | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

---

all\_expectation\_t Timing

First Click (1)

Last Click (2)

Page Submit (3)

Click Count (4)

---

Page Break



all\_comfort\_dev How would you feel if the developer of [\\${e://Field/most\\_familiar\\_app}](#) could read the following data from the cloud-based backups created by the app?

|                                                                                                      | Very<br>Uncomfortable<br>(1) | Somewhat<br>Uncomfortable<br>(2) | Neutral<br>(3)        | Somewhat<br>Comfortable<br>(4) | Very<br>Comfortable<br>(5) |
|------------------------------------------------------------------------------------------------------|------------------------------|----------------------------------|-----------------------|--------------------------------|----------------------------|
| The <b>name of the website</b> on which each account was created<br>(all_comfort_dev_website)        | <input type="radio"/>        | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/>          | <input type="radio"/>      |
| The <b>username</b> for each account<br>(all_comfort_dev_username)                                   | <input type="radio"/>        | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/>          | <input type="radio"/>      |
| The <b>secret</b> required to generate valid TOTP codes for each account<br>(all_comfort_dev_secret) | <input type="radio"/>        | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/>          | <input type="radio"/>      |



all\_comfort\_3rd How would you feel if other third parties (*not* the developer of [\\${e://Field/most\\_familiar\\_app}](#)) could read the following data from the cloud-based backups created by the app?

|                                                                                                      | Very<br>Uncomfortable<br>(1) | Somewhat<br>Uncomfortable<br>(2) | Neutral<br>(3)        | Somewhat<br>Comfortable<br>(4) | Very<br>Comfortable<br>(5) |
|------------------------------------------------------------------------------------------------------|------------------------------|----------------------------------|-----------------------|--------------------------------|----------------------------|
| The <b>name of the website</b> on which each account was created<br>(all_comfort_3rd_website)        | <input type="radio"/>        | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/>          | <input type="radio"/>      |
| The <b>username</b> for each account<br>(all_comfort_3rd_username)                                   | <input type="radio"/>        | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/>          | <input type="radio"/>      |
| The <b>secret</b> required to generate valid TOTP codes for each account<br>(all_comfort_3rd_secret) | <input type="radio"/>        | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/>          | <input type="radio"/>      |

all\_comfort\_t Timing

First Click (1)

Last Click (2)

Page Submit (3)

Click Count (4)

End of Block: Leaky Backups - Current Users

---

Start of Block: Account Lockout / Recovery - Part 1

### Q220 Account Lockout Questions

---



locked\_out How often have you been unable to provide a TOTP code within 5 minutes of being prompted to enter one when logging into **personal, non-work account(s)**?

☐ Never (1)

☐ Rarely (2)

☐ Sometimes (3)

☐ Frequently (4)

☐ Always (5)

---

locked\_out\_t Timing

First Click (1)

Last Click (2)

Page Submit (3)

Click Count (4)

End of Block: Account Lockout / Recovery - Part 1

---

Start of Block: Account Lockout / Recovery - Part 2



lockout\_reason Why were you unable to provide the TOTP code for **personal, non-work account(s)**? Select all that apply.

- ☐ My device was not immediately available, but I could have gotten a TOTP code if I chose to. (1)
  - ☐ I uninstalled the TOTP app from my device (2)
  - ☐ My device was lost (3)
  - ☐ My device was stolen (4)
  - ☐ My device was broken (5)
  - ☐ I purchased a new device and no longer had my old device (6)
  - ☐ Other... (7) \_\_\_\_\_
- 

lockout\_reason\_t Timing

First Click (1)  
Last Click (2)  
Page Submit (3)  
Click Count (4)

---

Page Break \_\_\_\_\_

Display this question:

If `lockout_reason = 2`

Or `lockout_reason = 3`

Or `lockout_reason = 4`

Or `lockout_reason = 5`

Or `lockout_reason = 6`

Or `lockout_reason = 7`



recovery\_within How long were you unable to login to your **personal, non-work account(s)** that required a TOTP code? Select all that apply.

- ☐ I successfully logged into some of those account(s) **within 1 hour** (1)
- ☐ I successfully logged into some of those account(s) **within 1 day** (2)
- ☐ I successfully logged into some of those account(s) **within 1 week** (3)
- ☐ I successfully logged into some of those account(s) **after more than 1 week** (4)
- ☐ I am **still unable** to login to some of those account(s) (5)

---


Page Break

Display this question:

If `lockout_reason = 2`  
 Or `lockout_reason = 3`  
 Or `lockout_reason = 4`  
 Or `lockout_reason = 5`  
 Or `lockout_reason = 6`  
 Or `lockout_reason = 7`



recovery\_how How did you try to login to those **personal, non-work account(s)**? Select all that apply.

- ☐ I contacted customer support for the website that I needed to login to. (1)
- ☐ I contacted the developer of the TOTP app for help. (2)
- ☐ I used the "Forgot Password" feature for the website that I needed to login to. (3)
- ☐ I used an account recovery code that I previously downloaded from the website that I needed to login to. (4)
- ☐ I installed the TOTP app on a new device and used the cloud-based backup feature that I previously enabled. (5)
- ☐ I had the website send me a login code via text message (SMS) when I needed to login. (6)
- ☐ I had the website send me a login code via email when I needed to login. (7)
- ☐ Other... (8) \_\_\_\_\_
- ☐  do not know (9)

recovery\_t Timing

First Click (1)

Last Click (2)

Page Submit (3)

Click Count (4)




Page Break

---



changed\_settings Which login settings did this experience cause you to change on those **personal, non-work account(s)**? Select all that apply.

- ☐ I changed the password on some or all of those accounts (1)
- ☐ I **temporarily** disabled TOTP on some or all of those accounts (2)
- ☐ I **permanently** disabled TOTP on some or all of those accounts (3)
- ☐ I enabled a different form of protection on some or all of those accounts. Please explain... (4) \_\_\_\_\_
- ☐  did not change any login settings (5)

---

changed\_settings\_t Timing  
First Click (1)  
Last Click (2)  
Page Submit (3)  
Click Count (4)

End of Block: Account Lockout / Recovery - Part 2

---


Start of Block: Demographic Questions

Q17.1 Demographic Questions

---



gender What is your gender?

- ☐ Woman (1)
- ☐ Man (2)
- ☐ Non-binary (3)
- ☐ Prefer to self-describe: (4)
- 
- ☐  decline to answer (5)



education What is the highest degree or level of school you have completed? (If you're currently enrolled in school, please indicate the highest degree you have *received*.)

- ☐ Less than a high school diploma (1)
- ☐ High school degree or equivalent (e.g. GED) (2)
- ☐ Some college, no degree (3)
- ☐ Associate degree (e.g. AA, AS) (4)
- ☐ Bachelor's degree (e.g. BA, BS) (5)
- ☐ Master's degree (e.g. MA, MS, MEd) (6)
- ☐ Professional degree (e.g. MD, DDS, DVM) (7)
- ☐ Doctorate (e.g. PhD, EdD) (8)
- ☐ I decline to answer (9)



background\_in\_tech Do you have a background in technology or computers?

☐ Yes (1)

☐ No (2)



tech\_savvy Please indicate how much you agree or disagree with the following statement:

|                                                 | Strongly<br>disagree (1) | Disagree (2)          | Neither agree<br>nor disagree<br>(3) | Agree (4)             | Strongly<br>Agree (5) |
|-------------------------------------------------|--------------------------|-----------------------|--------------------------------------|-----------------------|-----------------------|
| I consider<br>myself to be<br>tech savvy<br>(1) | <input type="radio"/>    | <input type="radio"/> | <input type="radio"/>                | <input type="radio"/> | <input type="radio"/> |



income What is your annual household income?

☐ Less than \$20,000 (1)

☐ \$20,000 to \$39,999 (2)

☐ \$40,000 to \$74,999 (3)

☐ \$75,000 to \$99,999 (5)

☐ \$100,000 to \$149,999 (6)

☐ \$150,000 or more (7)

☐ I decline to answer (8)



age How old are you?

- ☐ 18 to 29 (1)
  - ☐ 30 to 39 (2)
  - ☐ 40 to 49 (3)
  - ☐ 50 to 59 (4)
  - ☐ 60 to 69 (5)
  - ☐ 70 and above (7)
  - ☐ I decline to answer (8)
- 

demographic\_t Timing

First Click (1)  
Last Click (2)  
Page Submit (3)  
Click Count (4)

End of Block: Demographic Questions

---

Start of Block: Thanks, etc

the\_end **Thank you** Here are links to the studies that you may have seen referenced during the survey: [Security and Privacy Failures in Popular 2FA Apps](#) [Tweet about Google Authenticator's cloud-backup feature](#) Please message me in Prolific if any portions of this survey were confusing and/or problematic. Thanks, Conor Gilsean PhD Student

---

the\_end\_t Timing

First Click (1)  
Last Click (2)  
Page Submit (3)  
Click Count (4)

End of Block: Thanks, etc

---

### **B.3 Participant instructions to check whether cloud backups are enabled or disabled**

The pages of this section contain the instructions that were shown to participants in a Google Doc during the main survey when they were asked to check whether their TOTP app had cloud backups enabled or disabled.

Select your app and platform to  
view the instructions on how to  
determine whether cloud-based  
backups are enabled or disabled.

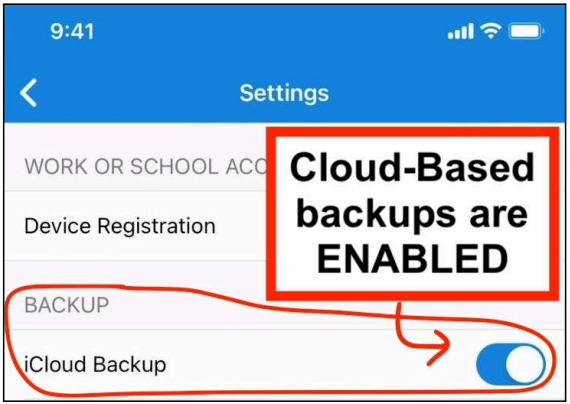
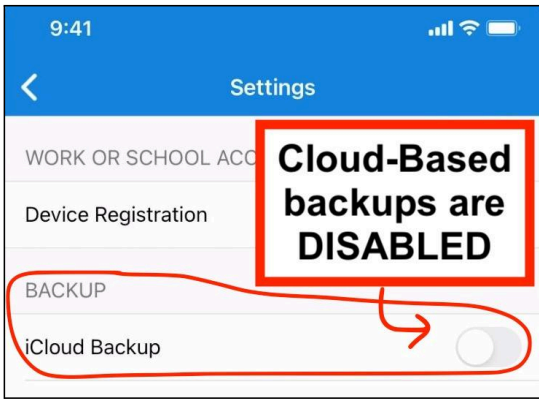
The attention word for the next survey question is listed at the top of each set of instructions.

|                                           |          |
|-------------------------------------------|----------|
| <u>Microsoft Authenticator on iOS</u>     | <u>2</u> |
| <u>Microsoft Authenticator on Android</u> | <u>3</u> |
| <u>Google Authenticator on iOS</u>        | <u>4</u> |
| <u>Google Authenticator on Android</u>    | <u>5</u> |
| <u>Duo Mobile on iOS</u>                  | <u>6</u> |
| <u>Duo Mobile on Android</u>              | <u>7</u> |
| <u>Authy on iOS</u>                       | <u>8</u> |
| <u>Authy on Android</u>                   | <u>9</u> |

## Microsoft Authenticator on iOS

The attention word for the next survey question is purple.

1. Open the menu in the top left of the home screen
2. Select **Settings** (gear icon)
3. The toggle labeled **iCloud Backup** indicates whether cloud-based backups are enabled or disabled. See the screenshots below for examples.

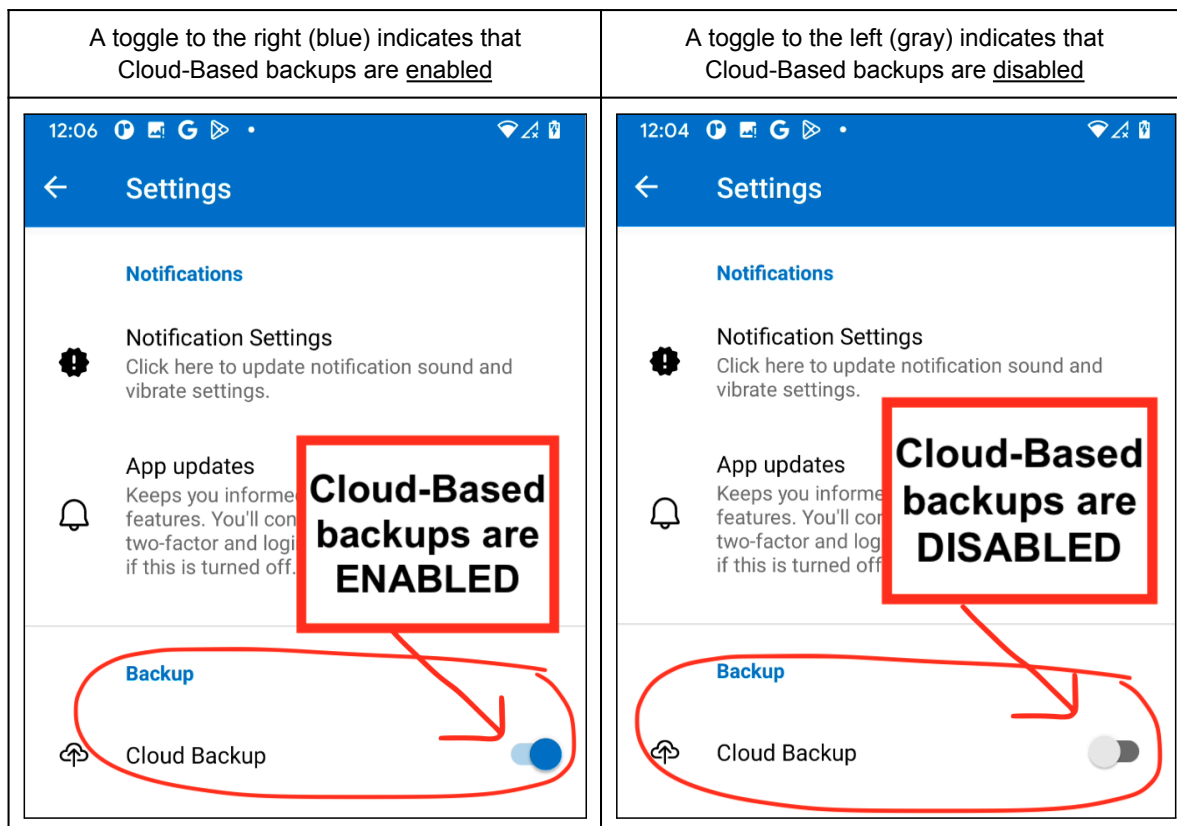
| A toggle to the right (blue) indicates that Cloud-Based backups are <u>enabled</u>                                                                                                                                                                                                                                                                     | A toggle to the left (gray) indicates that Cloud-Based backups are <u>disabled</u>                                                                                                                                                                                                                                                                       |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  <p>The screenshot shows the 'Settings' app on an iPhone. The 'BACKUP' section is highlighted with a red circle. A red arrow points from a red-bordered box containing the text 'Cloud-Based backups are ENABLED' to the blue toggle switch for 'iCloud Backup'.</p> |  <p>The screenshot shows the 'Settings' app on an iPhone. The 'BACKUP' section is highlighted with a red circle. A red arrow points from a red-bordered box containing the text 'Cloud-Based backups are DISABLED' to the gray toggle switch for 'iCloud Backup'.</p> |



## Microsoft Authenticator on Android

The attention word for the next survey question is purple.

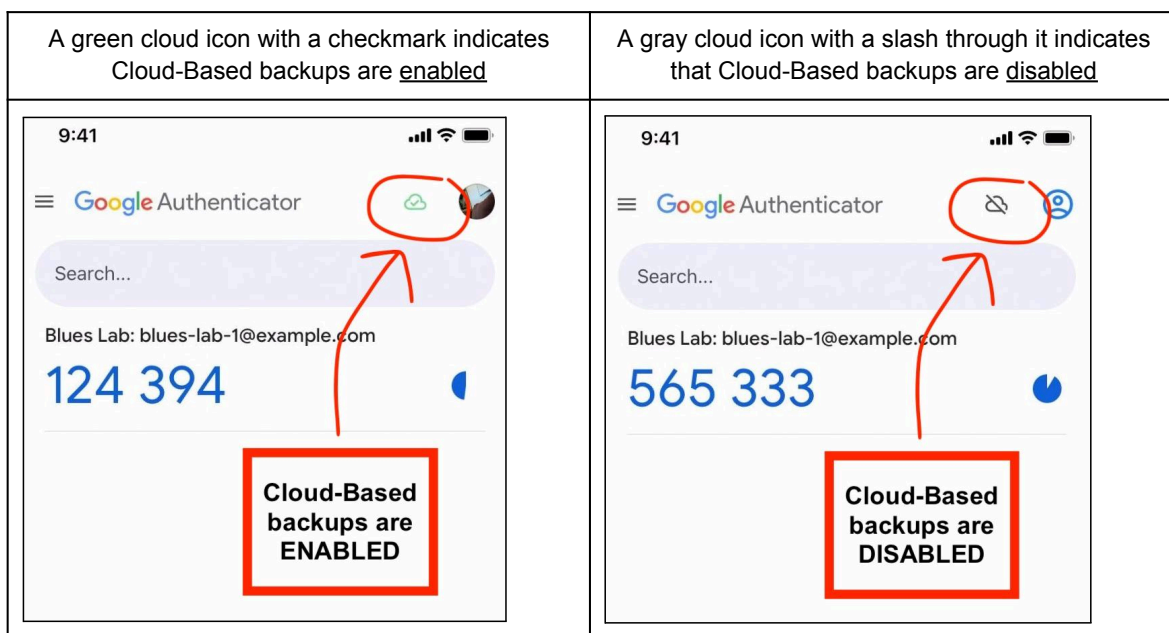
1. Open the menu in the top right of the home screen
2. Select **Settings** (gear icon)
3. The toggle labeled **Cloud Backup** indicates whether cloud-based backups are enabled or disabled. See the screenshots below for examples.



## Google Authenticator on iOS

The attention word for the next survey question is purple.

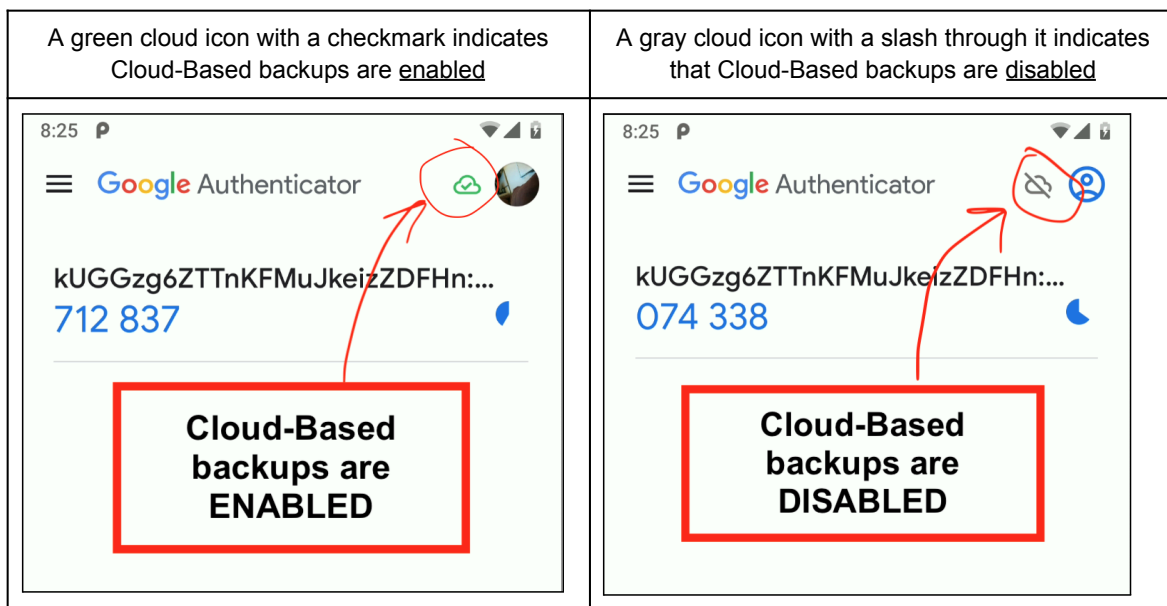
1. The cloud icon in the top right corner of the home screen indicates whether cloud-based backups are enabled or disabled. See the screenshots below for examples.



## Google Authenticator on Android

The attention word for the next survey question is purple.

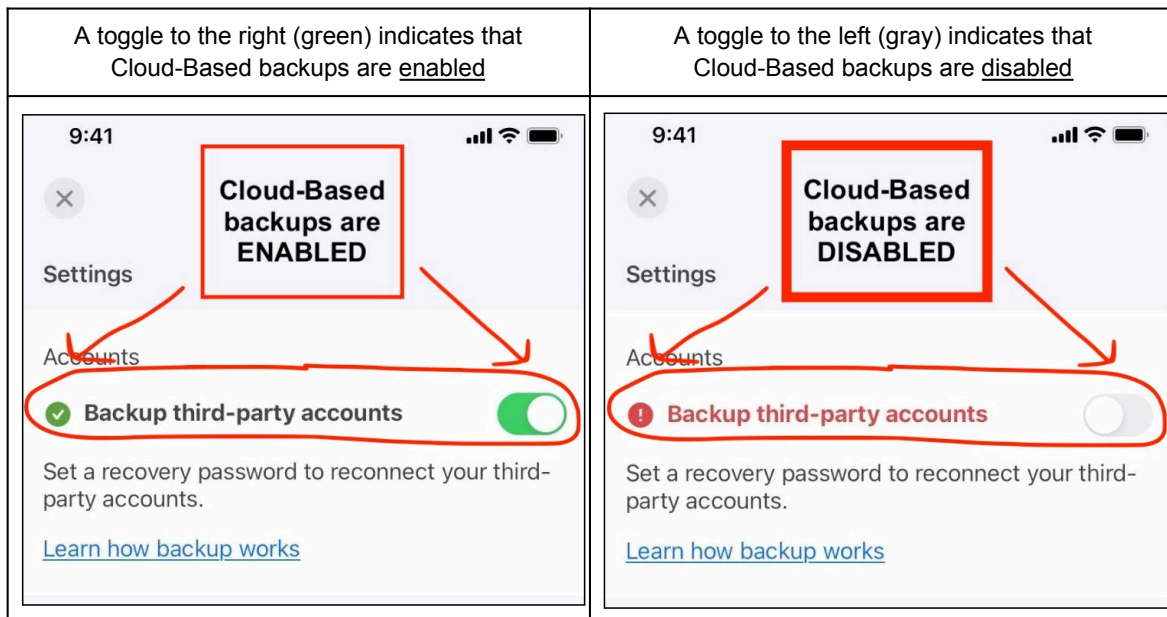
1. The cloud icon in the top right corner of the home screen indicates whether cloud-based backups are enabled or disabled. See the screenshots below for examples.



## Duo Mobile on iOS

The attention word for the next survey question is purple.

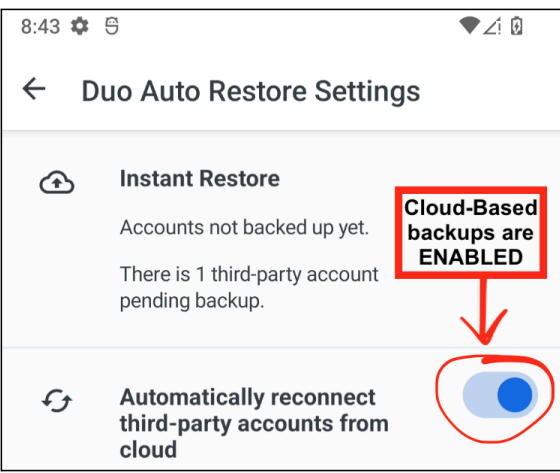
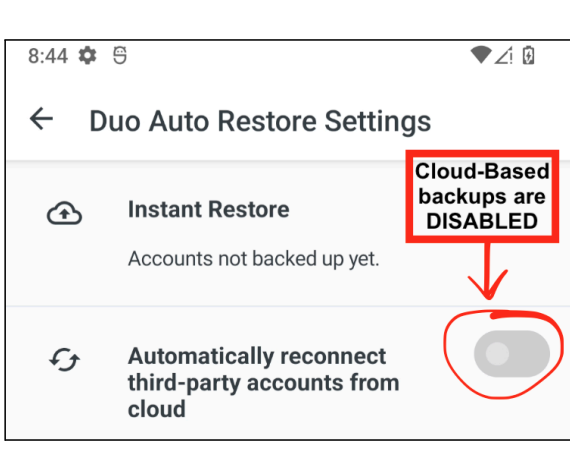
1. Open the menu in the top left of the home screen
2. Select **Settings** (gear icon)
3. The toggle labeled **Backup third-party accounts** indicates whether cloud-based backups are enabled or disabled. See the screenshots below for examples.



## Duo Mobile on Android

The attention word for the next survey question is [purple link](#).

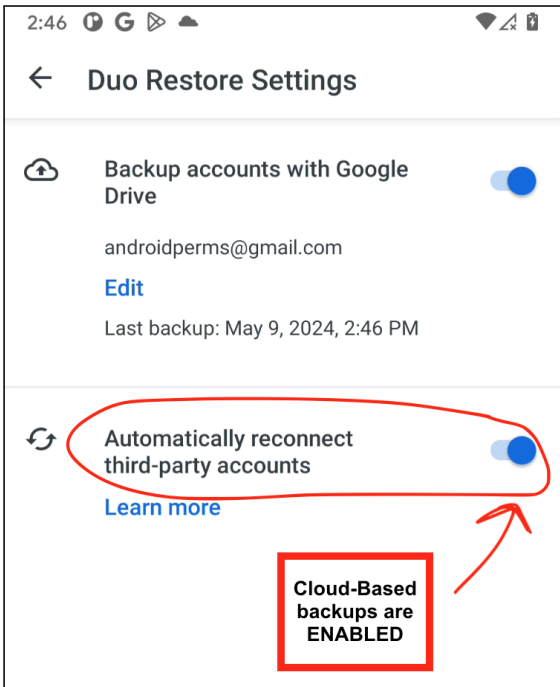
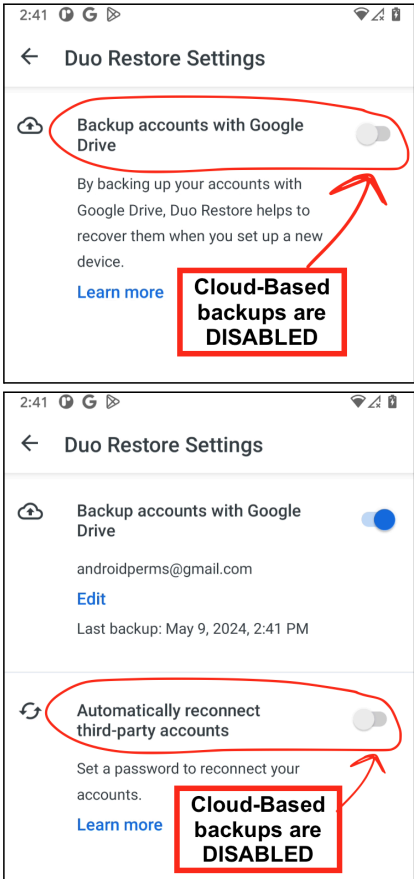
1. Open the menu in the top left of the home screen
2. Tap **Settings** (gear icon), then tap **Duo Instant Restore** (in the section titled General)
  - a. *Note: If you do not see an option for Duo Instant Restore, then you may have a legacy backup enabled. Please [follow the instructions here](#) for legacy backups.*
3. The toggle labeled **Automatically reconnect third-party accounts from cloud** indicates whether cloud-based backups are enabled or disabled. See the screenshots below for examples.

| A toggle to the right (blue) indicates that Cloud-Based backups are <u>enabled</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | A toggle to the left (gray) indicates that Cloud-Based backups are <u>disabled</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  <p>The screenshot shows the 'Duo Auto Restore Settings' screen. At the top, the time is 8:43. Below the title, there are two sections. The first section, 'Instant Restore', has a cloud icon and text: 'Accounts not backed up yet. There is 1 third-party account pending backup.' The second section, 'Automatically reconnect third-party accounts from cloud', has a circular arrow icon and a toggle switch that is turned on (blue). A red box highlights the toggle switch with the text 'Cloud-Based backups are ENABLED' and a red arrow pointing to it.</p> |  <p>The screenshot shows the 'Duo Auto Restore Settings' screen. At the top, the time is 8:44. Below the title, there are two sections. The first section, 'Instant Restore', has a cloud icon and text: 'Accounts not backed up yet.' The second section, 'Automatically reconnect third-party accounts from cloud', has a circular arrow icon and a toggle switch that is turned off (gray). A red box highlights the toggle switch with the text 'Cloud-Based backups are DISABLED' and a red arrow pointing to it.</p> |

## Duo Mobile on Android (legacy backups)

The attention word for the next survey question is purple.

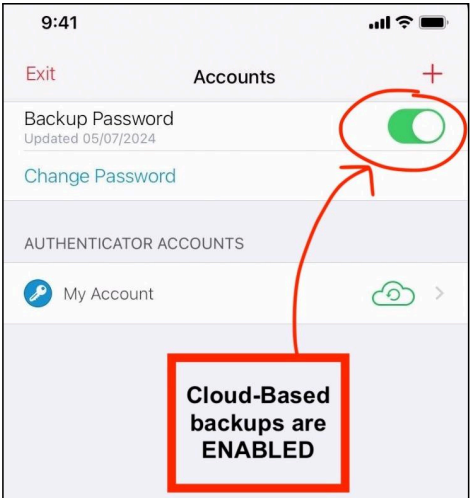
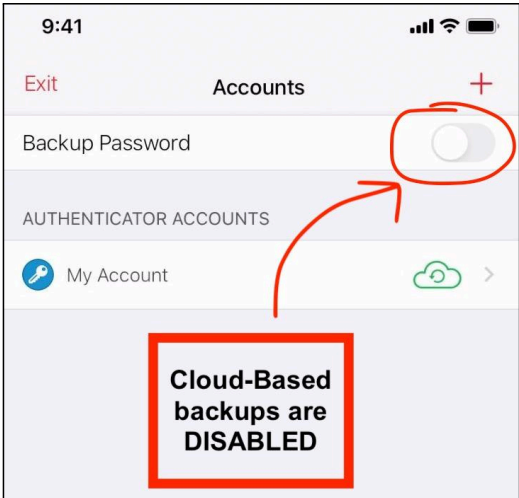
1. Open the menu in the top left of the home screen
2. Tap **Settings** (gear icon) and **Duo Restore** (in the section titled General)
3. The toggle labeled **Automatically reconnect third-party accounts** indicates whether cloud-based backups are enabled or disabled. See the screenshots below for examples.  
*Note:* The toggle for *Automatically reconnect third-party accounts* is only visible if the toggle for *Backup accounts with Google Drive* is enabled.

| A toggle to the right (blue) indicates that Cloud-Based backups are <u>enabled</u>                                                                                                                                                                                                                                                                                                                           | A toggle to the left (gray) indicates that Cloud-Based backups are <u>disabled</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  <p>2:46</p> <p>← Duo Restore Settings</p> <p>Backup accounts with Google Drive</p> <p>androidperms@gmail.com</p> <p><a href="#">Edit</a></p> <p>Last backup: May 9, 2024, 2:46 PM</p> <p>Automatically reconnect third-party accounts</p> <p><a href="#">Learn more</a></p> <p><b>Cloud-Based backups are ENABLED</b></p> |  <p>2:41</p> <p>← Duo Restore Settings</p> <p>Backup accounts with Google Drive</p> <p>By backing up your accounts with Google Drive, Duo Restore helps to recover them when you set up a new device.</p> <p><a href="#">Learn more</a></p> <p><b>Cloud-Based backups are DISABLED</b></p> <p>2:41</p> <p>← Duo Restore Settings</p> <p>Backup accounts with Google Drive</p> <p>androidperms@gmail.com</p> <p><a href="#">Edit</a></p> <p>Last backup: May 9, 2024, 2:41 PM</p> <p>Automatically reconnect third-party accounts</p> <p>Set a password to reconnect your accounts.</p> <p><a href="#">Learn more</a></p> <p><b>Cloud-Based backups are DISABLED</b></p> |

## Authy on iOS

The attention word for the next survey question is purple.

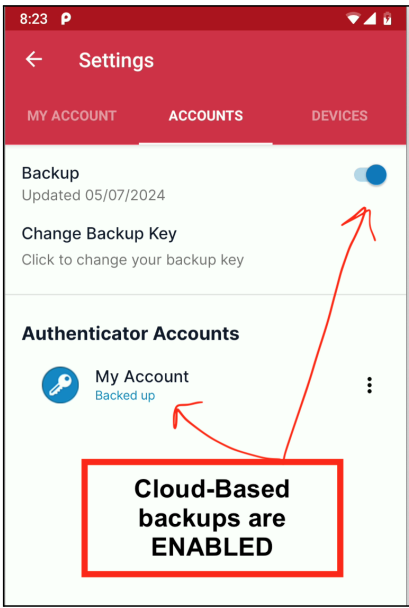
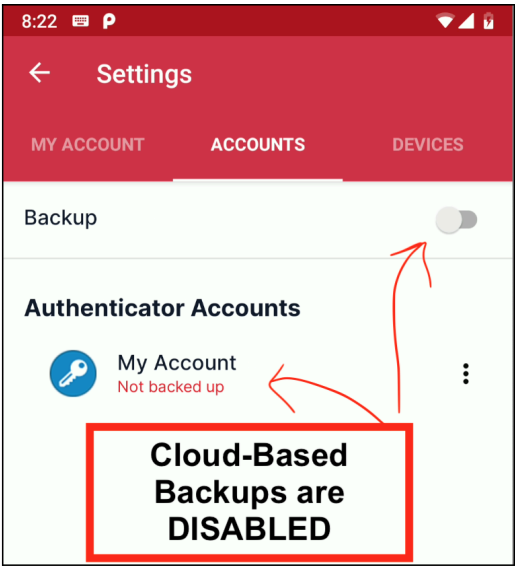
1. Open the settings menu by pressing the gear icon in the upper right corner of the home screen
2. In the settings menu, select the Accounts tab in the center bottom menu
3. The toggle labeled **Backup Password** indicates whether cloud-based backups are enabled or disabled. See the screenshots below for examples.

| A toggle to the right (green) indicates that Cloud-Based backups are <u>enabled</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | A toggle to the left (gray) indicates that Cloud-Based backups are <u>disabled</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  <p>The screenshot shows the 'Accounts' settings page. At the top, there's a status bar with the time 9:41 and signal/battery icons. Below that, a red 'Exit' button and a '+' icon are visible. The 'Accounts' title is centered. The 'Backup Password' section shows 'Updated 05/07/2024' and a green toggle switch. Below this is a 'Change Password' link. The 'AUTHENTICATOR ACCOUNTS' section lists 'My Account' with a cloud icon and a right arrow. A red box at the bottom contains the text 'Cloud-Based backups are ENABLED'. A red arrow points from this box to the green toggle switch.</p> |  <p>The screenshot shows the 'Accounts' settings page. At the top, there's a status bar with the time 9:41 and signal/battery icons. Below that, a red 'Exit' button and a '+' icon are visible. The 'Accounts' title is centered. The 'Backup Password' section shows a gray toggle switch. Below this is a 'Change Password' link. The 'AUTHENTICATOR ACCOUNTS' section lists 'My Account' with a cloud icon and a right arrow. A red box at the bottom contains the text 'Cloud-Based backups are DISABLED'. A red arrow points from this box to the gray toggle switch.</p> |

## Authy on Android

The attention word for the next survey question is purple.

1. Open the settings menu by tapping the 3-dots icon in the upper right corner of the home screen and selecting **Settings**.
2. In the settings menu, select the **Accounts** tab in the center top menu bar.
3. The toggle labeled **Backup** indicates whether cloud-based backups are enabled or disabled. See the screenshots below for examples.

| A toggle to the right (blue) indicates that Cloud-Based backups are <u>enabled</u>                                                                                                                                                                                                                                                                                                                                                                                                             | A toggle to the left (gray) indicates that Cloud-Based backups are <u>disabled</u>                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  <p>The screenshot shows the 'Settings' screen with the 'ACCOUNTS' tab selected. The 'Backup' toggle is turned on (blue). Below it, the 'Authenticator Accounts' section shows 'My Account' with a blue key icon and the text 'Backed up'. A red box at the bottom contains the text 'Cloud-Based backups are ENABLED'. Red arrows point from the box to the 'Backup' toggle and the 'My Account' entry.</p> |  <p>The screenshot shows the 'Settings' screen with the 'ACCOUNTS' tab selected. The 'Backup' toggle is turned off (gray). Below it, the 'Authenticator Accounts' section shows 'My Account' with a blue key icon and the text 'Not backed up'. A red box at the bottom contains the text 'Cloud-Based Backups are DISABLED'. Red arrows point from the box to the 'Backup' toggle and the 'My Account' entry.</p> |