BlueTOTP: Designing Phishing-Resistant and User-Friendly Two-Factor Authentication

Marian Käsemodel[1][0009−0007−1267−2751],
Verena Winterhalter[2][0000−0003−0752−3480], Felix Heisel[1][0009−0004−6677−3484],
Florian Alt[2][0000−0001−8354−2195], and Bastian Pfleging[1][0000−0003−0505−9338]

[1] TU Bergakademie Freiberg, Germany
[2] LMU Munich, Germany

Abstract. Two-factor authentication (2FA) is an effective measure to safeguard against password attacks (e.g., guessing, credential stuffing). However, many existing 2FA methods are neither user-friendly nor protect adequately against phishing, particularly real-time (person-in-the-middle) attacks. We introduce *BlueTOTP*, a novel approach that leverages Bluetooth to automatically transmit time-based one-time passwords (TOTP) and verify the requesting domain before issuing the second factor. By reducing manual interactions and ensuring domain legitimacy, BlueTOTP streamlines the authentication process and mitigates phishing risks. We present the design and implementation of BlueTOTP, followed by an evaluation of its usability and performance. BlueTOTP not only improves the user experience during authentication but also significantly reduces the overall time required to complete 2FA authentication.

1 Introduction

Knowledge-based authentication, such as passwords, remains a common method to protect user accounts. However, easy-to-guess passwords, as well as the widespread reuse of weak passwords, leave accounts susceptible to brute force and credential-stuffing attacks. In response, two-factor authentication (2FA) emerged as a more robust defense, typically requiring users to provide an additional factor beyond their password. A popular method is time-based one-time passwords (TOTP), commonly generated on a separate device such as a smartphone.

Despite TOTP's security benefits, two key challenges persist: *usability* and *phishing resistance*. While many 2FA tools, such as Google Authenticator, are usable, manually entering TOTPs can be cumbersome and time-consuming, increasing user burden and the likelihood of errors. Moreover, traditional TOTP methods cannot verify that the requesting website is legitimate. Attackers can exploit this by relaying users' credentials and TOTP codes through a fake website, enabling real-time phishing (man-in-the-middle) attacks [16].

To address this, we introduce *BlueTOTP*, an enhanced TOTP-based 2FA system that leverages Bluetooth Low Energy (BLE) to automate TOTP entry and verify the requesting domain. BlueTOTP transfers the TOTP from the user's smartphone to the browser without manual input, reducing the cognitive load and time needed for authentication. Simultaneously, it checks whether the requesting domain matches the legitimate site, mitigating phishing risks.

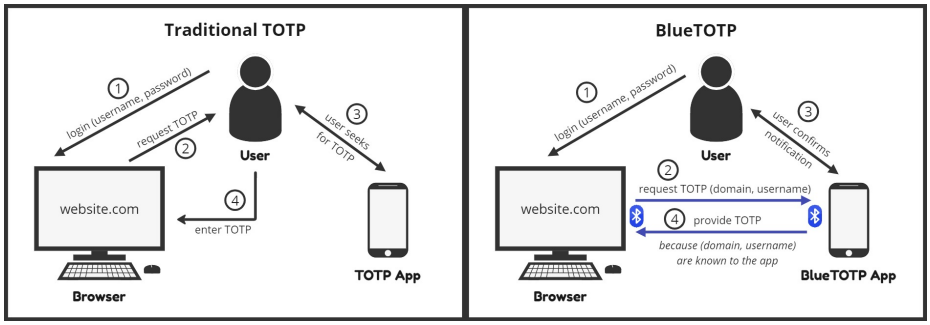


Fig. 1. BlueTOTP improves the security of TOTP-based two-factor authentication by preventing person-in-the-middle attacks. Connecting the browser and TOTP phone app speeds up interaction and suppresses submitting codes to unauthorized websites.

We developed a prototype to assess the effectiveness of our concept. Results from our user study show that BlueTOTP provides better usability and user experience than traditional TOTP approaches. Consistent with predictions from a Keystroke-Level Model (KLM) analysis, BlueTOTP shortens the authentication process by 42–52%. Through automated code transfer and domain verification, BlueTOTP thus offers a faster and more secure TOTP-based 2FA procedure.

Contribution Statement. Our contributions are technical and empirical in nature [18]: (1) We present an enhanced approach to TOTP-based two-factor authentication, along with a proof-of-concept implementation. (2) Through insights gathered from a qualitative user study, we demonstrate how our method positively affects usability and user experience. (3) We provide design implications for human-centered security researchers.

2 Background and Related Work

We begin by introducing key terminology and technical background on two-factor authentication (2FA), followed by a review of prior work.

2.1 Terminology

Two-Factor Authentication (2FA) verifies an individual's identity using two distinct factors. There are three broad categories of authentication factors [11]: knowledge of information ("something you know"), possession of an object ("something you have"), and inherence of a unique characteristic ("something you are"). Most commonly, passwords (i.e., knowledge-based credentials) are used to authenticate users. However, passwords are susceptible to multiple risks: anyone with access to the password can impersonate the user, and weak, user-chosen passwords can often be guessed or brute-forced [2]. To increase security, 2FA combines at least two factors, reducing vulnerabilities inherent to

single-factor (password-only) authentication. As a subset of multi-factor authentication (MFA), 2FA adds an extra layer of defense, making attacks significantly harder.

Real-Time Phishing (a form of man-in-the-middle attack) occurs when attackers lure users to a deceptive website that mimics a legitimate one [16]. Unaware of the fraud, users enter their login credentials (and possibly TOTP codes), which the attacker then relays in real time to the legitimate website. 2FA methods verifying the authenticity of the visited domain offer effective protection.

2.2 Approaches to 2FA

Reese et al. [13] provide an overview of the most prevalent 2FA methods, which typically revolve around an additional element owned by the user.

Pre-Generated Codes Users receive a list of single-use codes to be entered after password login. While simple, these codes are prone to loss, theft, and entry errors, making them impractical for regular use.

One-Time Passwords (OTP) Many services send temporary codes via SMS or email after login. This avoids extra hardware but remains vulnerable to interception (e.g., SIM swapping [9, 6]) and phishing. In the EU, *smsTAN* is now banned for banking due to these risks.

Time-Based One-Time Password (TOTP) TOTP uses a shared secret on the user's device to generate codes every 30 seconds without needing a network. It avoids SIM-based attacks but requires opening an app and manually entering the code—often a cumbersome process.

Push Notifications Some 2FA methods send login prompts via push notifications. While convenient, this can lead to "prompt bombing" or "MFA fatigue"[3], where users approve requests without verifying them.

FIDO/Universal Second Factor (U2F) U2F uses physical security keys (e.g., YubiKey[4]) or special software. It is secure against man-in-the-middle attacks but requires users to carry a separate device. A mobile-based alternative [12] using Bluetooth was proposed but lacked usability evaluation.

FIDO2/Passkeys Passkeys rely on biometrics and can sync via cloud services. Though more user-friendly, they raise privacy concerns (e.g., iCloud storage), can fail (e.g., due to injury), and depend on hardware and website support. Thus, they are likely to complement rather than replace existing 2FA methods.

2.3 Prior Research on (the Usability of) 2-Factor Authentication

Although hardware-based methods like FIDO tokens provide strong phishing protection, many users avoid them due to the hassle of carrying a separate

[3] https://sosafe-awareness.com/de/glossar/mfa-fatigue-angriff/
[4] https://www.yubico.com/der-yubikey/, last access: 2024-09-12

device. Meanwhile, many websites still rely on 2FA methods vulnerable to real-time phishing or offer no secure alternatives. As a result, the main threat has shifted from brute force to sophisticated, real-time phishing attacks.

Shirvanian and Agrawal [14] address the usability gap with their "2D-2FA" approach, wherein users draw a swipe pattern on a smartphone app—similar to Android's unlock method—which generates a PIN for the server. While this method is innovative in substituting swipe patterns for numerical codes, it does not inherently guard against phishing, as attackers can relay the generated PIN.

A more phishing-resistant approach is *PhotoAuth* by Sun et al. [15]. After logging in, the user receives a link on their phone, takes a photo of the PC browser's address bar, and the server verifies the domain via image recognition. While PhotoAuth blocks real-time phishing (attackers can't forward the photo), it faces challenges like typosquatting[5]. Its usability has yet to be fully evaluated.

Commercial solutions, like the fee-based 1Password[6], store TOTP secrets on their own servers and automatically fill in TOTP codes for users. While this is convenient, it also creates a single point of compromise: if 1Password is breached, both the user's password and second factor could be exposed.

Most relevant to our work is *2FA Phishing Prevention* (2FA-PP) by Ulqinaku et al. [17], which uses the Web Bluetooth API to link the user's browser with their smartphone via BLE. A challenge-response protocol verifies the URL and blocks man-in-the-middle (MITM) phishing—if latency allows. However, its automatic approval mode poses a risk: an attacker with physical access and a stored password can log in if the user is nearby, implicitly consenting to 2FA. While 2FA-PP avoids browser modifications, questions about user acceptance and everyday usability remain open.

In contrast, our proposed *BlueTOTP* solution requires explicit confirmation on the user's phone, thus preventing "behind-the-back" approvals. We advocate for integrating this functionality directly into future browsers to verify URLs and prevent real-time phishing—without imposing burdens on website operators.

Finally, Colnago's study [4] highlighted how newcomers often describe 2FA as "easy" once they get used to it and value its security benefits. Positive initial experiences, in turn, can encourage users to adopt 2FA for more accounts. We aim to make 2FA methods even more accessible, driving long-term adoption.

3 Research Problem & Concept

Login procedures for passwords combined with TOTP generally follow a straightforward pattern: after entering their username and password, the user is prompted to open a TOTP app (often on a smartphone) and manually copy the displayed code to the website. While simple, this workflow faces two primary challenges:

Usability /User Experience Manually copying TOTP codes from one device to another increases cognitive load and time spent, especially when users must

[5] Typosquatting = Exploiting visually similar characters (e.g., lowercase l" and uppercase I") depending on the browser's font rendering

[6] https://support.1password.com/one-time-passwords/, last access: 2024-09-12

juggle multiple devices. Typographical errors are common, causing failed login attempts or necessitating re-entry. Since each TOTP is valid for only about 30 seconds, users may feel rushed to enter the code in time or else wait for a fresh one, compounding both stress and the chance of error.

Real-Time Phishing Current TOTP implementations do not verify the domain where users enter the code. Consequently, attackers can create pixel-perfect phishing sites to intercept both the password and TOTP in real-time. Because the TOTP app is unaware of which domain requested the code, it cannot detect fraudulent sites, thus enabling person-in-the-middle attacks.

3.1 Design Goals and Conceptual/Technical Contribution

BlueTOTP augments the traditional TOTP workflow with a secure, Bluetooth-based channel between the user's browser and smartphone. This design enhances both security and usability while preserving the familiar TOTP infrastructure used by many websites. Specifically, BlueTOTP addresses three key objectives:

Security By exchanging vital information (e.g., the current domain) between the browser and the smartphone app, BlueTOTP ensures that the one-time password is released only if the requested domain matches the legitimate site. This mechanism dramatically lowers the risk of real-time phishing attacks, as users cannot inadvertently provide TOTPs to malicious pages. In addition, an active confirmation on the user's phone prevents "behind-the-back" approvals.

Usability & User Experience Rather than requiring users to open a TOTP app and manually type a rapidly expiring code, BlueTOTP automatically transfers the TOTP from the phone to the appropriate field in the browser. This automation alleviates time pressure, reduces the chance of errors (e.g., typos), and curtails the mental burden on users. It effectively streamlines the login flow and fosters a more positive 2FA experience.

Integration & Deployment BlueTOTP integrates with existing TOTP infrastructures without necessitating any changes on the part of website operators. Instead, it relies on a secure and trusted Bluetooth connection between the user's smartphone and browser, which can be standardized or embedded in future browser releases. This approach simplifies deployment and accelerates adoption, all while maintaining both security and usability benefits.

3.2 BlueTOTP Concept

Unlike typical TOTP methods that require manual code entry, *BlueTOTP* creates a secure, real-time connection between the browser and the smartphone's TOTP app (e.g., via Bluetooth). While our long-term goal is browser-level integration, our prototype shows this can be achieved via a browser extension. We focus on how BlueTOTP affects everyday usability and UX, emphasizing the authentication workflow over the setup process. BlueTOTP users follow 5 steps:

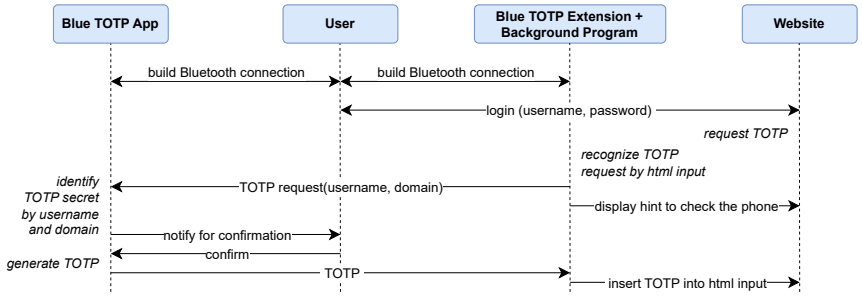


Fig. 2. BlueTOTP login process

1. **Username and Password Entry.** The user navigates to a 2FA-protected website and enters their username and password.
2. **TOTP Prompt.** After verifying first-factor credentials, the website prompts for a TOTP code, as in conventional 2FA. The browser detects that 2FA is enabled, extracts the domain from the URL, and recalls the username from the login form. An optional HTML attribute on the TOTP field can aid detection and auto-completion.
3. **Secure Bluetooth Query.** The browser sends the domain and username to the phone's TOTP app via a secure, pre-paired Bluetooth channel. (Encryption and connection details are set during initial pairing.)
4. **App Validation and Approval.**
   (a) If the TOTP app recognizes the domain and username, the user receives a smartphone notification requesting approval to share the TOTP. Upon confirmation, the app securely returns the current TOTP to the browser.
   (b) If the domain-username pair is unknown, the app and browser warn the user about a potential phishing site.
5. **Automatic Code Entry.** Upon receiving the TOTP, the browser auto-fills the field and submits the second factor. Similar to push-based 2FA, this avoids "MFA fatigue" by requiring both Bluetooth proximity and approval.

3.3 Usability and User Experience

From a usability standpoint, BlueTOTP offers several clear advantages over traditional TOTP solutions:

- **Reduced Errors.** Automatically transferring the TOTP to the browser eliminates typos and manual input mistakes common in traditional TOTP.
- **Faster Authentication.** Instead of hunting for a TOTP app on the smartphone, users swipe to confirm a notification. This streamlined process cuts down on both cognitive load and time spent. We believe that shorter authentication times reduce fatigue and improve user satisfaction as users do not have to interrupt their original activity for lengthy authentication.

– **Less Time Pressure.** Traditional TOTPs expire every 30 seconds, creating pressure to enter codes quickly or wait for the next cycle. BlueTOTP bypasses this countdown anxiety, providing a smoother experience.

While users still need to interact with their smartphone for explicit TOTP approval, the process is much simpler than opening a separate app and manually typing a code. In the future, an even more convenient approach could involve a smartwatch to display the same notification and allow the user to confirm the code via a quick tap or swipe—avoiding the need to retrieve the phone at all.

3.4 Usability Analysis: Comparing Task Completion Times

To quantify how much BlueTOTP can accelerate the login process compared to traditional TOTP methods, we performed a static Keystroke-Level Model (KLM) analysis [3]. KLM offers a structured way to estimate the time users spend on low-level actions (e.g., keystrokes, mouse moves, or taps). Although commonly applied to desktop interactions, we extend it to smartphone usage by incorporating operators for mobile and touchscreen interactions [8, 10].

Appendix 4 summarizes the KLM operators used. When users switch devices (e.g., from keyboard to phone) or shift visual focus, we assume these actions occur in parallel and record only the longer event. Hand transitions are treated like mouse-keyboard homing. This lets us approximate the time cost of each step in both traditional TOTP and our BlueTOTP workflow.

To streamline our KLM analysis, we focus solely on the time needed to acquire and confirm the second factor. We simplify placement of *Mental Act* operators and assume the following baseline conditions:

– The user is already on the TOTP input page, with hands on or above the keyboard. The phone rests within easy reach (we use 1.1 s for the I operator).
– The phone is secured with a lock pattern, taking appr. 3.0 s to unlock [7].
– After unlocking, the phone shows the last used app, so the user must navigate to the home screen to open the TOTP app.
– The TOTP app displays the current 6 digit code on launch, with no need to scroll or switch views.

Under these assumptions, our analysis (see Appendix, Table 1) estimates a task completion time (TCT) of 8 s with *BlueTOTP*, compared to 14 s using a traditional TOTP workflow—a reduction of approximately 43 %. If biometric authentication shortens phone unlocking to 0.5 s, the TCT for BlueTOTP further decreases to 5.5 s, compared to 11.5 s for traditional TOTP, yielding a 52 % improvement.

3.5 Security Analysis: Mitigating Real-Time Phishing and Defending Against Local Browser Access Attacks

As illustrated in Figure 3, *BlueTOTP* effectively mitigates real-time phishing attacks. In a typical phishing scenario, attackers trick users into entering their

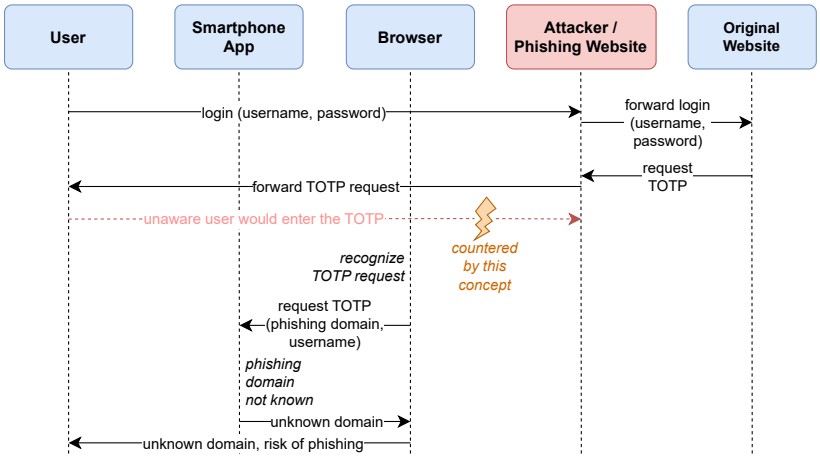


Fig. 3. How to counter real-time phishing

credentials on a fraudulent site. These credentials are then forwarded to the legitimate service, and the subsequent TOTP request is relayed to the fake site. Under traditional TOTP schemes, unsuspecting users would unwittingly provide the necessary code, granting the attacker full access.

In contrast, BlueTOTP ensures the browser recognizes domains linked to BlueTOTP-registered 2FA. Only then does it request a TOTP from the phone via a secure Bluetooth link[7]. The request includes a domain and username; if they don't match the data on the phone, the TOTP is withheld; e.g., a phishing site with the correct username but an unknown domain would trigger an alert.

Security Against Physical Access Attacks A common weakness in solutions like 2FA-PP [17] is exposure to nearby adversaries with physical access to an unlocked computer. If the attacker also has the user's credentials, they can authenticate with the second factor if the phone is within Bluetooth range. BlueTOTP counters this by requiring explicit approval on the user's phone before sending the TOTP to the browser, preventing "behind-the-back" authorizations.

Fallback Mechanism If the Bluetooth connection is unavailable, BlueTOTP offers a fallback. Scanning a QR code that contains the encrypted combination of domain and user name, the smartphone app continues to verify the domain-username pair locally but presents the TOTP for manual entry. This preserves protection against phishing, even without an active Bluetooth link, by ensuring the TOTP is generated only for trusted domain-username combinations.

[7] For real-world use, the Bluetooth channel must be secured with authentication and end-to-end encryption. Our prototype does not yet implement secure Bluetooth.

3.6 Research Approach

We followed a multi-stage approach to develop and evaluate *BlueTOTP*. First, we designed the core system—a Bluetooth-based enhancement to TOTP—to improve phishing resistance and everyday usability. We then ran a multi-day field study where participants used BlueTOTP in realistic settings, enabling a comparison to standard TOTP regarding security, speed, and UX.

4 BlueTOTP: Proof-of-Concept Implementation

Our *BlueTOTP* prototype uses Bluetooth Low Energy (BLE) to enhance both usability and security in TOTP-based 2FA. It has two main components: an Android smartphone app and a browser extension, which cooperate to automate TOTP handling without relying on native browser features.

Smartphone Application Based on the open-source *FreeOTP+*[8], the Android app manages TOTP secrets, generates codes, and handles BLE communication with the browser extension. During setup, users scan a QR code with the TOTP secret, domain, and username. The app then advertises as a BLE peripheral for discovery by the browser (acting as BLE central). A foreground service keeps it active for incoming requests. On login, the app checks the domain-username pair and sends a TOTP only if they match. Despite its underlying complexity, the interface remains simple for managing multiple TOTP entries.

Browser Extension A dedicated Chrome extension handles BLE communication, sends TOTP requests, and auto-fills the code in the website's form. Due to Chrome's Web Bluetooth limitations, an Electron-based background service manages scanning and connections.

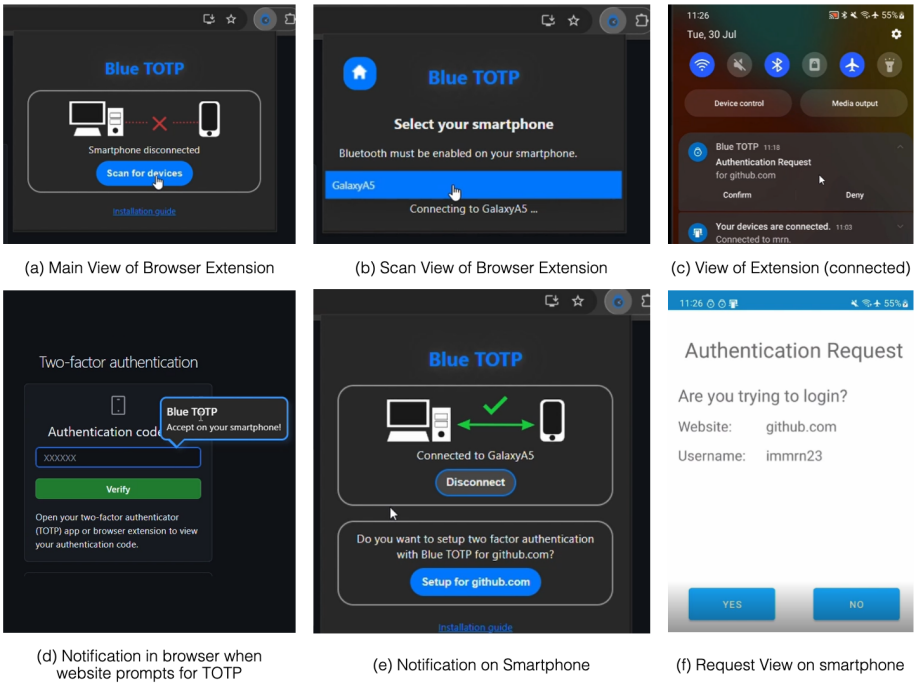
Workflow and Authentication Before login, the extension scans for the phone and establishes a BLE link. When the user enters credentials on a 2FA site, it detects the TOTP prompt and sends the domain and username to the phone. If they match a known entry, the phone requests user approval and returns the code—blocking malicious sites from obtaining a valid TOTP.

Electron-Based Background Service To bypass browser BLE restrictions, an Electron app runs in the system tray, managing scans, pairing, and data transfer. As browser support improves, this layer could become obsolete.

As shown in Figures 2 and 4, once connected, the user logs in as usual. The extension captures the TOTP prompt, sends the domain-username pair, and auto-fills the confirmed code—eliminating manual entry and reducing risk.

Setup Process To begin using *BlueTOTP*, the user must perform a one-time setup. First, the website generates a QR code for the user, who is then prompted

[8] https://github.com/helloworld1/FreeOTPPlus



(a) Main View of Browser Extension (b) Scan View of Browser Extension (c) View of Extension (connected)

(d) Notification in browser when website prompts for TOTP (e) Notification on Smartphone (f) Request View on smartphone

Fig. 4. Screenshots of the BlueTOTP browser extension and smartphone app. (a) The user opens the extension and connects to the BlueTOTP app. (b) They select their device from nearby options. (c) The main view confirms the connection. (d) After entering credentials on a website, the extension displays a prompt, asking the user to check their phone. (e) The smartphone shows a notification that can be quickly confirmed or denied. (f) Opening the notification reveals which user is requesting access.

to provide a one-time password. The user launches the smartphone app, which guides them through establishing a Bluetooth connection with the browser extension. Next, within the browser extension (on the relevant website tab), the user initiates the setup procedure; the extension offers hints for locating or displaying the QR code, which some sites may not automatically reveal. The user then enters or confirms their username in the extension, which can be pre-filled if previously saved. At this point, the framework is ready to associate the smartphone app with the website, ensuring that future TOTP generation and verification proceed seamlessly via the secure Bluetooth link.

5 Field Study: Assessing Users' Perception of BlueTOTP

We conducted a multi-day field study supported by interviews and questionnaires, approved by the authors' institutional ethics board.

The study aimed to answer the following question: How do users perceive the efficiency, user experience, and overall usability of the new *BlueTOTP* concept?

We hypothesized that *BlueTOTP* would (1) provide a more user-friendly alternative to traditional TOTP workflows, and (2) reduce the time needed for 2FA. The study simulated real-world usage to assess these aspects.

5.1 Apparatus

Our *BlueTOTP* prototype used three main components:

1. the smartphone app that generates and stores TOTP secrets,
2. the browser extension (including a background service) that manages TOTP operations on websites, and
3. a simulated banking web application functioning as an authentication testbed.

Participants were asked to log in to this banking app daily, providing us with a practical environment to observe *BlueTOTP* in action. To assess usability and user experience, we employed the System Usability Scale (SUS) and the User Experience Questionnaire (UEQ). We also collected demographic information and participants' prior experiences with TOTP-based 2FA via a brief survey. After the field study, we conducted semi-structured interviews to probe users' opinions, document any issues they faced, and explore how their attitudes toward 2FA evolved (see Appendix, Section 9.1).

5.2 Study Design

We conducted a multi-day field study, supplemented by interviews and questionnaires, to compare *BlueTOTP* against traditional TOTP. Our independent variable was the authentication method, and our key measures included the time needed for daily logins (recorded by a simulated banking app), plus SUS and UEQ ratings for the BlueTOTP setup (questionnaires completed after the introduction session) and daily login interactions (questionnaires completed after the end of the field study phase during the final interview), as well as for previous login experiences with traditional TOTP (also completed during the final interview). For the SUS, we inverted the negative items, so that a high value represented a positive evaluation, and then calculated the SUS by adding the values and multiplying them by 2.5, to get a final value between 0 and 100. For the UEQ, we calculated the mean values for each of the six dimensions *attractiveness, perspicuity, efficiency, dependability, stimulation and novelty.*

5.3 Procedure

We conducted our study in three phases, ensuring a balance between thorough data collection and minimal participant burden:

Introduction Session Participants signed consent forms and completed a demographic survey on prior TOTP usage. They then installed the BlueTOTP extension and app, logged into a simulated banking site, and completed the setup. Setup time was recorded without time pressure. Afterwards, participants completed the SUS and UEQ for the setup process and received €15.

Six-Day Field Usage For six days, participants logged into the banking site once daily. This environment simulated a security-critical transaction flow and allowed us to collect TOTP-entry times. Reminders were sent at 1 a.m. and 7 p.m. if participants had not yet logged in, and those who missed a day made two logins the next day. This happened twice during the study, leading to two participants having to perform two logins on the following day. Participants could also note any issues or observations on using BlueTOTP.

Final Interviews After the six-day period, we held brief, semi-structured interviews to gather overall impressions and discuss challenges. Participants then completed SUS and UEQ, comparing their recalled experiences with traditional TOTP to their actual use of BlueTOTP. Upon finalizing these assessments, they received an additional €25.

5.4 Recruitment and Participants

We recruited a total of 11 participants through a mailing list, requiring that all individuals had prior experience with TOTP-based authentication to ensure basic familiarity with two-factor setup and usage. Each participant needed a smartphone running at least Android 10.0 and a Windows 10 or Windows 11 computer with Bluetooth capabilities. Because the study materials (including the mobile app and the simulated banking interface) were in the authors' local language, participants also had to demonstrate sufficient language proficiency. Across the introduction session, daily logins, and the final interview, each participant invested appr. 2–3 hours and was compensated with a total of €40.

Of the 11 participants, five identified as male and six as female. Eight were students, two held full-time employment, and one was retired. Eight participants ranged between 20 and 30 years, two were between 51 and 65, and one was 18.

5.5 Data Analysis

We adopted an affinity diagram approach to interpret participants' qualitative feedback. Two researchers independently coded the interview transcripts, organizing key themes as notes on a digital whiteboard. A third researcher then merged overlapping topics to highlight distinct themes.

One participant's login data was excluded because they never successfully connected the browser extension and relied solely on the fallback mechanism, effectively bypassing the main BlueTOTP workflow.

6 Results

We outline the findings from our field study, focusing primarily on how participants perceived and used *BlueTOTP*. We divide our discussion into feedback on the overall concept and insights specific to the prototype implementation.

6.1 Assessment of the BlueTOTP Concept

Security, Usability, & User Experience Initially, we asked participants for their general *opinions on 2FA*. Half viewed it positively, citing stronger security as a key benefit, though four still found it "tedious." Two held neutral views. After introducing BlueTOTP's anti-phishing and time-saving features, six participants prioritized security over speed or usability, three focused more on usability or speed, and one assigned equal importance to all three aspects.

Opinions on login *task completion time* were divided. Four noted that establishing a Bluetooth connection could slow things down, especially if connectivity issues arose. Meanwhile, six found BlueTOTP faster overall, thanks to automatic TOTP transfers. Two participants were undecided. Our measurements showed a median TOTP entry time of 8.5 s—slightly above our KLM-based estimate (8 s), presumably due to real-world factors such as smartphone locking.

Regarding *usability and user experience*, most participants reported favorable impressions of BlueTOTP, aligning with the SUS and UEQ metrics. BlueTOTP earned an average SUS of 79 (vs. 73 for traditional TOTP), though the difference was not statistically significant ($z = -1.19$, $p = 0.235$). UEQ scores, however, did reveal significant gains for BlueTOTP on *attractiveness* ($t(9) = 4.432$, $p = 0.002$), *dependability* ($t(9) = 2.409$, $p = 0.039$), *stimulation* ($t(9) = 3.498$, $p = 0.007$), and *novelty* ($t(9) = 5.218$, $p < 0.001$).

Most participants felt that any extra setup effort was offset by the time and convenience benefits of day-to-day use, a sentiment reflected in the progressively shorter login durations. Besides efficiency gains, many also noted the improved phishing protection. Still, one participant found the setup too tedious, and another was hesitant to use 2FA at all despite its perceived advantages.

Overall Likes and Dislikes Many participants liked the automatic code transfer, noting it removed the need to watch for code expiration or retype codes. Once configured, confirming logins on the phone felt quick and simple. However, several participants found that manually enabling Bluetooth—if disabled by default—could slow them down.

After six days, half of the participants reported a more favorable view of 2FA, citing improved efficiency and ease of use. Others remained unchanged in their opinions. Almost all favored BlueTOTP over traditional 2FA but would still only opt into extra authentication for critical accounts if given a choice. In cases where 2FA was mandatory, six participants said they would be more willing to comply if BlueTOTP were offered, whereas two still resisted 2FA altogether.

6.2 Prototype

Some comments addressed the *prototype* implementation itself, emphasizing how actual workflows can diverge from initial design intentions.

Login Procedure Participants used different strategies when authenticating with BlueTOTP. Four began by opening the browser extension, scanning for Bluetooth devices, and pairing their phone before entering credentials. They then approved the TOTP notification, which auto-filled the website's form. Six others opened the BlueTOTP app first and then initiated scanning via the extension, believing it yielded a faster or more reliable connection. In theory, manual app launching is unnecessary, since the Android service runs in the background.

Despite the simplicity, five participants reported confusion about the correct sequence, sometimes entering credentials before pairing. One considered resetting BlueTOTP for a specific service, doubting they had set it up properly. These insights underline the need for clear instructions on pairing order and login steps, helping users feel confident about when to connect devices and enter credentials.

Bluetooth Half of the participants felt scanning for devices took longer than anticipated, though once connected, reliability was good. Only one incident arose in which the app falsely showed an active connection. While most participants (8/10) left Bluetooth on continuously for their computers, they typically disabled it on smartphones until needed. Two kept Bluetooth always on for both devices.

Technical Aspects When asked about the BlueTOTP app running in the background, seven participants noticed it, primarily because of the permanent notification. Of these, three occasionally stopped the app due to battery concerns or annoyance with the notification. No one observed significant battery drain, though two commented that the constant notification was somewhat irritating. One participant estimated the app's battery use at only 1–2% during the study.

Participants reported a few minor glitches. In three cases, the browser extension duplicated the phone's entry while scanning. One participant experienced a situation where the app showed a "connected" status but never pushed the TOTP notification. Another worried about what would happen if they lost their phone, pointing to the need for a reliable fallback or recovery option.

Suggestions for Improvement Seven participants wanted a more robust, automated Bluetooth pairing process, hoping to avoid manual scans. Three also suggested bypassing the extra "Submit" click by auto-submitting the TOTP after transfer. Additional ideas included allowing device pairing *after* login credentials are entered, a way to resend the TOTP if notification is missed, and prompting users if they attempt to set up BlueTOTP on an account already protected by 2FA. Some participants requested interface refinements, such as a light-mode theme or larger, clearer status indicators in the app. Although none tested smartwatch integration, two expressed interest in confirming TOTP requests on a watch, further reducing the need to handle the phone directly.

7 Discussion

7.1 Usefulness & Usability of the Concept

BlueTOTP addresses core usability hurdles in TOTP-based 2FA by automating code entry. Rather than searching for a TOTP app, unlocking it, and typing in codes, users simply confirm a prompt on their phone. This reduces manual effort, lowers error rates, and counters the common critique that 2FA is cumbersome.

Comparison to Traditional TOTP We recruited participants familiar with TOTP, enabling direct comparisons with prior habits. Our KLM analysis indicates that BlueTOTP can reduce TOTP entry times by 43–52%, and empirical data support these gains. Over six days, median entry times dropped from 9,s to 5,s, aligning with KLM predictions (5.5–8,s, depending on unlocking). These results echo Reese et al.[13], who reported a 15.1,s median for traditional TOTP—close to our 14,s benchmark. In contrast, BlueTOTP reached a median of 8.5,s, outperforming even the best results in their study.

SUS scores and UEQ scales for *perspicuity* and *efficiency* showed no significant difference, while *attractiveness*, *dependability*, *stimulation*, and *novelty* were statistically significant. We attribute this to the novelty of BlueTOTP's Bluetooth-based interaction, which offers a new experience in the TOTP context. The lack of significance in efficiency-related measures may reflect user uncertainty about whether BlueTOTP is truly faster.

Learning Effects and Automatic Reconnection Although not statistically significant, participants felt that login speed improved over time. Six of ten believed BlueTOTP outperformed conventional TOTP, and two found it similar. Some pointed out slow Bluetooth initiation, which did not impact measured TOTP entry times as the timer started *after* connection. Implementing automatic reconnection would remove the need for manual pairing each time.

Manual Confirmation vs. Fully Automated TOTP A more automated scheme could skip notification confirmations [17], but that raises risks if attackers gain physical access to an already "trusted" computer. Maintaining a manual approval step in BlueTOTP preserves user control and prevents such attacks.

7.2 Usefulness & Usability of the Concept

BlueTOTP tackles long-standing usability challenges in TOTP-based 2FA by automating code entry. Instead of locating a TOTP app, switching views, and carefully typing codes, users simply confirm a prompt on their phone. This substantially lightens the cognitive load and reduces the potential for input errors, addressing criticisms that TOTP can be inconvenient for day-to-day use.

Comparison to Traditional TOTP Our decision to recruit participants already familiar with TOTP allowed them to scrutinize how BlueTOTP differs from established 2FA practices. Both the Keystroke-Level Model (KLM) projections and our empirical data suggest considerable efficiency gains. In particular, login times dropped from around nine seconds on the first day to approximately five seconds on the last, aligning with KLM estimates (5.5–8 s, depending on phone unlocking). These findings resonate with Reese et al. [13], whose participants needed a median of 15.1 s for traditional TOTP, reinforcing that manual code entry is both time-consuming and prone to small, compounding inefficiencies. We propose a follow-up study involving simulated phishing to explore user behavior under attack. The design of BlueTOTP can thus demonstrate its strengths in the field of phishing protection (verifying domain matches and preventing TOTP release on mismatch).

Learning Effects and Automatic Reconnection Although our correlation analysis did not find a significant relationship between days of usage and TOTP entry time, participants themselves consistently reported feeling faster over time. Some identified the initial Bluetooth connection as a bottleneck, though it did not affect our measured TOTP entry time. Automatic reconnection could eliminate the manual pairing step, further smoothing the authentication experience.

Manual Confirmation vs. Full Automation Skipping smartphone confirmation could speed up 2FA further [17], but removing manual approval introduces risks—especially if an attacker gains access to a trusted computer. BlueTOTP retains user confirmation to keep individuals in control of the second factor. This reflects the trade-off between convenience and essential security checks. Automating code transfer improves usability, as our data show, while preserving user involvement in critical decisions.

HCI Contribution Our primary contribution to HCI lies in the integration of domain-checking in addition to the TOTP mechanisms – both within a user-centered framework, evaluated using established usability metrics such as SUS and UEQ. We show the potential for future HCI frameworks to combine phishing resistance mechanisms with UX assessments, illustrating how minimal user interaction can effectively enhance security without compromising usability.

7.3 Incentivizing Users to Enable Bluetooth Permanently

For a seamless 2FA experience, users must keep Bluetooth enabled on both phones and computers. Initially, some participants disabled Bluetooth due to privacy, security, or battery worries [1], yet our findings show that tangible benefits—such as faster logins—often outweigh those concerns. Two participants left Bluetooth on for their computers and one on their phones.

Bluetooth Low Energy significantly limits power drain [1], and Bluetooth 6.0 promises even lower consumption. None of our participants reported notable battery issues. The move toward always-on Bluetooth is evident in wearables,

smart-home devices, and recent Android updates, which allows passive scanning by default. This supports the persistent connectivity *BlueTOTP* relies on.

Privacy concerns remain valid, though MAC randomization and BLE fingerprint obfuscation [5] help mitigate tracking risks. Seamless pairing and rapid 2FA confirmations can justify enabling Bluetooth by default, provided users understand these privacy safeguards. Educating users about such enhancements can help them confidently adopt continuous Bluetooth usage.

7.4 How to Convince People to Use More Usable and Secure 2FA

BlueTOTP minimizes manual steps compared to traditional TOTP, but this can obscure how it shields users from phishing. Our results show that once participants realized *BlueTOTP* checks the domain on the phone, six of ten ranked security above usability. Emphasizing that the smartphone independently validates a site's legitimacy—rather than relying on the user—increases trust. This clarity can motivate wider 2FA adoption by underscoring not only BlueTOTP's convenience but also its capacity to block phishing attempts effortlessly.

7.5 Limitations and Constraints

BlueTOTP currently cannot be used on devices without Bluetooth or where it's disabled (e.g., by corporate policies). Bluetooth was chosen to ensure local presence of the second factor, but future versions could support WiFi or VPN connections. The fallback mechanism could also be extended to handle both setup and authentication. As a proof of concept, the current prototype does not yet include full cryptographic hardening.

As our recruiting primarily targeted a university population, most participants were students. Together with the small sample, this may constrain the broader applicability of our findings. Future work could include a larger, more diverse group of participants to improve the generalizability of our findings.

Although we captured user interactions within the simulated banking application, we could not reliably distinguish between logins completed via the BlueTOTP workflow and logins achieved through the fallback mechanism. Participants generally reported using the fallback option infrequently, but at least six of the 59 valid logins employed it. Since fallback authentications likely took longer, any time measurements for BlueTOTP may be overestimated.

Furthermore, participants were asked to recall their past usage of traditional TOTP methods, which they did not use during the study. This reliance on memory, coupled with the absence of a parallel control group, poses constraints on interpreting the differences between traditional and BlueTOTP experiences.

8 Future Work

Our findings suggest multiple directions for improving and expanding *BlueTOTP*. Below, we highlight key avenues for both practitioners and researchers.

8.1 Toward Standardization and Browser Integration

Although the *BlueTOTP* prototype demonstrates core benefits, its long-term potential lies in native browser integration. Such standardization would reduce reliance on external extensions and enhance trust by embedding 2FA directly into the browser. A standardized *BlueTOTP* could offer:

- *Automated Discovery and Secure Connections.* Browsers would automatically detect *BlueTOTP* apps over Bluetooth, using a secure protocol (e.g., TLS) to maintain authenticated, encrypted links with auto-reconnect.
- *Initial Pairing.* Users would authorize pairing once, after which the browser and phone exchange cryptographic keys for seamless future connections.
- *Auto-Confirmation of TOTP.* Following TOTP injection, the browser could confirm the prompt without additional clicks, further speeding logins.
- *Guided 2FA Setup.* Recognizing HTML tags for TOTP fields would let browsers guide users through setup, generating and syncing secrets.
- *Uniform User Experience.* Native browser support would standardize the interface across devices and OSes, lowering confusion and driving adoption.

Both our prototype and any future standard depend on clear HTML attributes (e.g., `autocomplete="one-time-code"`, `id="totp"`, `name="totp"`) to identify TOTP fields. Aligning with such emerging web standards can deliver a consistent, user-friendly defense against phishing and other 2FA threats.

8.2 Additional Security Considerations Regarding Phishing

When the phone does not recognize a requested domain-username pair, *BlueTOTP* must handle three possible situations: (1) an active phishing attack, (2) a legitimate website originally set up via a standard TOTP app, or (3) a legitimate setup process mistakenly triggering *BlueTOTP*. Since *BlueTOTP* cannot distinguish these automatically, it should caution users that phishing may be occurring, but also clarify the benign alternatives. Future work could explore UI designs that help users interpret such warnings more accurately.

If the phone and browser cannot connect via Bluetooth, *BlueTOTP* provides a fallback. One proposed method displays a short hash of the domain and username on the browser, which the user enters into their phone to verify legitimacy before revealing the TOTP. This maintains phishing protection, but a poor UI could confuse users. Studies should pinpoint interface strategies that effectively guide people through warnings, fallback prompts, and additional steps.

8.3 Directions for Further Studies

Future work should refine and test phishing alerts in *BlueTOTP* and similar systems. A multi-day field approach could periodically expose participants to pixel-perfect phishing sites (or phishing emails) without prior warning, revealing spontaneous reactions and how often users disregard domain-based alerts.

Researchers could also compare UX when setting up TOTP via a traditional app versus *BlueTOTP*, examining which approach better helps users spot phishing attempts. Another key scenario involves accidental TOTP setup during an intended login, prompting investigations into how clearly the system distinguishes setup from login. Studying these nuances in a natural, longitudinal setting can guide refinements that strengthen *BlueTOTP* and broader anti-phishing efforts. The effects of the lack of time pressure due to BlueTOTP should be statistically investigated in future studies.

9 Conclusion

We introduce *BlueTOTP*, an extension of conventional TOTP-based 2FA that establishes a secure, trusted channel between the user's browser and smartphone. By exchanging the browser's domain information with the phone, BlueTOTP prevents real-time (man-in-the-middle) phishing attacks. Simultaneously, the trusted connection automates TOTP transfer from the smartphone to the web form, lowering task completion times and enhancing usability and UX.

References

1. Barua, A., Al Alamin, M.A., Hossain, M.S., Hossain, E.: Security and privacy threats for bluetooth low energy in iot and wearable devices: A comprehensive survey. IEEE Open Journal of the Communications Society **3**, 251–281 (2022). https://doi.org/10.1109/OJCOMS.2022.3149732
2. Bošnjak, L., Sreš, J., Brumen, B.: Brute-force and dictionary attack on hashed real-world passwords. In: 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). pp. 1161–1166 (2018). https://doi.org/10.23919/MIPRO.2018.8400211
3. Card, S.K., Moran, T.P., Newell, A.: The keystroke-level model for user performance time with interactive systems. Commun. ACM **23**(7), 396–410 (jul 1980). https://doi.org/10.1145/358886.358895, https://doi.org/10.1145/358886.358895
4. Colnago, J., Devlin, S., Oates, M., Swoopes, C., Bauer, L., Cranor, L., Christin, N.: "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. p. 1–11. CHI '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3173574.3174030, https://doi.org/10.1145/3173574.3174030
5. Givehchian, H., Bhaskar, N., Redding, A., Zhao, H., Schulman, A., Bharadia, D.: Practical obfuscation of ble physical-layer fingerprints on mobile devices. In: 2024 IEEE Symposium on Security and Privacy (SP). pp. 2867–2885 (2024). https://doi.org/10.1109/SP54263.2024.00073
6. Hallman, R.A.: Sim swapping attacks for digital identity theft: A threat to financial services and beyond (2023)
7. Harbach, M., von Zezschwitz, E., Fichtner, A., Luca, A.D., Smith, M.: It's a hard lock life: A field study of smartphone (Un)Locking behavior and risk perception. In: 10th Symposium On Usable Privacy and Security (SOUPS 2014). pp. 213–230. USENIX Association, Menlo Park, CA (Jul 2014), https://www.usenix.org/conference/soups2014/proceedings/presentation/harbach

8. Holleis, P., Otto, F., Hussmann, H., Schmidt, A.: Keystroke-level model for advanced mobile phone interaction. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. p. 1505–1514. CHI '07, Association for Computing Machinery, New York, NY, USA (2007). https://doi.org/10.1145/1240624.1240851, https://doi.org/10.1145/1240624.1240851
9. Jover, R.P.: Security analysis of sms as a second factor of authentication. Commun. ACM **63**(12), 46–52 (nov 2020). https://doi.org/10.1145/3424260, https://doi.org/10.1145/3424260
10. Lee, A., Song, K., Ryu, H.B., Kim, J., Kwon, G.: Fingerstroke time estimates for touchscreen-based mobile gaming interaction. Human Movement Science **44**, 211–224 (2015). https://doi.org/10.1016/j.humov.2015.09.003, https://www.sciencedirect.com/science/article/pii/S0167945715300373
11. Marky, K., Ragozin, K., Chernyshov, G., Matviienko, A., Schmitz, M., Mühlhäuser, M., Eghtebas, C., Kunze, K.: "nah, it's just annoying!" a deep dive into user perceptions of two-factor authentication. ACM Trans. Comput.-Hum. Interact. **29**(5) (oct 2022). https://doi.org/10.1145/3503514, https://doi.org/10.1145/3503514
12. Rathi, S., Oswal, H., D'Souza, A., Pratham, S., Khedkar, V.: Enhancing user experience and integrating passwordless fido-based authentication through bluetooth low energy (ble). In: 2024 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS). pp. 1–5 (2024). https://doi.org/10.1109/ICBDS61829.2024.10837342
13. Reese, K., Smith, T., Dutson, J., Armknecht, J., Cameron, J., Seamons, K.: A usability study of five Two-Factor authentication methods. In: Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019). pp. 357–370. USENIX Association, Santa Clara, CA (Aug 2019), https://www.usenix.org/conference/soups2019/presentation/reese
14. Shirvanian, M., Agrawal, S.: 2d-2fa: A new dimension in two-factor authentication. In: Proceedings of the 37th Annual Computer Security Applications Conference. p. 482–496. ACSAC '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3485832.3485910, https://doi.org/10.1145/3485832.3485910
15. Sun, Y., Zhu, S., Zhao, Y., Sun, P.: A user-friendly two-factor authentication method against real-time phishing attacks. In: 2022 IEEE Conference on Communications and Network Security (CNS). pp. 91–99. IEEE (2022). https://doi.org/10.1109/CNS56114.2022.9947253
16. Ulqinaku, E., Assal, H., Abdou, A., Chiasson, S., Capkun, S.: Is real-time phishing eliminated with {FIDO}? social engineering downgrade attacks against {FIDO} protocols. In: 30th USENIX Security Symposium (USENIX Security 21). pp. 3811–3828 (2021), https://www.usenix.org/conference/usenixsecurity21/presentation/ulqinaku
17. Ulqinaku, E., Lain, D., Capkun, S.: 2fa-pp: 2nd factor phishing prevention. In: Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks. p. 60–70. WiSec '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3317549.3323404, https://doi.org/10.1145/3317549.3323404
18. Wobbrock, J.O.: Seven research contributions in hci. Intelligence **174**(12-13), 910–950 (2012)

Appendix

9.1 Questions of the Final Interview

1. Did you encounter any problems within the field study?
2. You had the task of logging into the website and performing a simulated transaction for 6 days. How was your experience using BlueTOTP? Did you like anything, or did something bother you?
   (a) *If negative:* Why did X bother you, and would this prevent you from using BlueTOTP?
   (b) What would need to change for you to enjoy using BlueTOTP?
3. Did you note any (other) problems that occurred during the study?
4. What was your attitude toward 2FA before the study? Has using BlueTOTP changed your view of 2FA (and how)?
5. Imagine you can use 2FA with a service. To what extent would you be willing to use 2FA with BlueTOTP?
6. Imagine you are required to use 2FA with a service (e.g., your employer or the service itself requires it). To what extent would you accept this requirement if you could use BlueTOTP?
7. The goal of BlueTOTP is to make 2FA with time-based one-time passwords faster and more user-friendly. Did you feel that BlueTOTP was slower or faster during login compared to the traditional process (and how)?
8. *Briefly explain why it is ideally faster (the app doesn't need to be searched for on the smartphone since BlueTOTP sends a notification, and the one-time password doesn't need to be read and entered in the browser since it happens automatically). Briefly explain the additional protection it offers (for phishing websites, BlueTOTP does not send a notification since it cannot generate one-time passwords for phishing sites).* Which feature is most important to you: enhanced security, user-friendliness, faster handling, or none of these?
9. Regarding BlueTOTP: How do you view the balance between the effort required to set it up and the actual benefit during login?
   (a) *Possible follow-up:* Does the benefit during login (faster, more user-friendly) outweigh the extra effort during setup?
10. What steps did you follow when you had to log in and authenticate? (For example, first opened the app, then connected the extension with the app, then logged into the website, etc.)
11. The BlueTOTP app continues to run in the background even when you close it (e.g., swipe it away in the task manager). Were you aware of this?
12. You can also stop this background process of BlueTOTP (Task Manager → Foreground Services → BlueTOTP → Stop). How often did you stop the background process, and if so, why?
13. Did the connection between the Android app and the browser extension work smoothly, or were there any issues in between? What issues?
14. What is the usual status of your computer's Bluetooth function? (Mostly on, only turned on when needed) Has the use of BlueTOTP changed this behavior?

15. What is the usual status of your smartphone's Bluetooth function? (Mostly on, only turned on when needed) Has the use of BlueTOTP changed this behavior?
16. In your perception, did your smartphone's battery drain faster than usual during the 6-day usage period?
17. Did you use a smartwatch to confirm the notification to log into the website?
    (a) If yes, do you find it comfortable to use BlueTOTP this way?
18. Which method would you prefer to use: the traditional TOTP method or BlueTOTP? Why?
19. Where do you see advantages and disadvantages in BlueTOTP?
20. Do you have any other comments or suggestions for improvements to BlueTOTP?

Table 1. Comparison of interaction sequences and task completion times of BlueTOTP vs. a traditional TOTP login

BlueTOTP			**Traditional TOTP**		
Operator	Time (s)	Explanation	Operator	Time (s)	Explanation
R(t)	0,50	Browser sends TOTP request to phone, phone shows notification	H	0,40	Move hand to mouse
			P_d	1,10	Point cursor to TOTP input field
			BB	0,20	Click to activate field
I	1,10	Pick up phone in front of the user	I	1,10	Pick up phone in front of the user
S_{macro}	0,36	direct visual attention to phone screen	S_{macro}	0,36	direct visual attention to phone screen
A(t)	3,00	Activate & unlock phone [7]	A(t)	3,00	Activate & unlock phone [7]
F	0,12	Flick from top to reveal notifications	P_s	0,43	Tap on on home button to redirect to home screen
P_s	0,43	Tap on accept button to confirm sharing TOTP with browser	M	1,35	Find TOTP App icon
R(t)	0,50	App returns TOTP to Browser	P_s	0,43	Tap on TOTP app icon to start App
			R(t)	0,50	TOTP app starting time
			M	1,35	locate TOTP for current login process and remember the 6 digits
			H, S_macro	0,40	move one hand from phone to computer keyboard
			6*K	1,20	type the six TOTP digits
			K	0,20	hit ENTER to submit the TOTP form
R(t)	2,00	Browser autocompletes TOTP, submits form and returns successfully with protected website	R(t)	2,00	Browser returns successfully with protected website
	8,01	**Task completion time using BlueTOTP**		**14,02**	**Task completion time using traditional TOTP**

Table 2. 2FA time needed with BlueTOTP, median per day

Day	1	2	3	4	5	6
median [s]	9.2	10.9	7.3	7.9	7.3	5.2

Table 3. TOTP entry timings of traditional TOTP [13] and BlueTOTP

Method	Q1	Median	Q3
Traditional TOTP	10.7 s	15.1 s	23.3 s
BlueTOTP	5.5 s	8.5 s	15.8 s

Table 4. Overview of the KLM operators used in our analysis

Operator	**Explanation**	**Time**	**Device**	**Source**
K, Keystroke	Typing a (computer) key (assumption: average skilled typist)	0.2 s	desktop	[3]
P_d, Pointing	Point to target with mouse	1.1 s	desktop	[3]
BB, Clicking	Mouse button click	0.2 s	desktop	[3] (adapted)
M, Mental act	Subsumed time for mental acts	1.35 s	desktop	[3, 8]
H, Homing	Move hand between mouse and keyboard	0.4 s	desktop	[3]
$R(t)$, System Response Time	Time needed by a system to respond to user input (as long as it blocks the user from subsequent steps)	variable	any	[3]
I, Initial Act	Initial act of locating the phone (e.g., from the desk). Can be longer incase located in a pocket or elsewhere (on average 4.6 s)	1.1 s	phone	[8]
S_{Macro}, Macro Attention Shift	Shift focus between display and real world (or other display)	0.36 s	phone	[8]
$A(t)$, Action	Time to perform certain complex action with phone (variable)	variable	phone	[8]
T, Tapping	(Repetitive) touch on a closely located target area	0.31 s	smart phone	[10]
P_s, Pointing	Touch on a target area that requires a larger finger movement	0.43 s	smart phone	[10]
F, Flicking	Quick swipe/flick gesture on touch screen with arbitrary length	0.12 s	smart phone	[10]
D, Dragging	Dragging an object on the touch screen	0.17 s	smart phone	[10]