Anchors of Trust: A Usability Study on User Awareness, Consent, and Control in Cross-Device Authentication

Xin Zhang
Fudan University
zhangx22@m.fudan.edu.cn

Xiaohan Zhang
Fudan University
xh_zhang@fudan.edu.cn

Huijun Zhou
Fudan University
zhouhj24@m.fudan.edu.cn

Bo Zhao
Fudan University
bzhao23@m.fudan.edu.cn

***Abstract*—Cross-device authentication (*XDAuth*) has become an essential mechanism for seamless account access across multiple devices. In this paradigm, a user can sign in on one device (the *target device*) by completing authentication on another trusted device (the *authentication device*) that holds an active session or stored credentials, improving user experience. However, the decoupling of the authentication device and target device introduces new risks: the physical and contextual separation disrupts the usual authentication flow, creates information asymmetry, and makes it hard for users to assess the legitimacy of an authentication request. Consequently, users may inadvertently approve malicious logins and face account compromise, especially when key contextual details, explicit confirmation, or revocation mechanisms are missing.**

To address these risks, we start from a user-centric perspective grounded in three fundamental user rights: the *right to know*, the *right to consent*, and the *right to control*, to safeguard the security and usability of *XDAuth* systems. We investigate how these rights are supported in practice by examining 27 major services spanning three typical *XDAuth* schemes. Our findings are concerning: over half of the services do not provide any information about the target device during authentication, not all services enforce explicit user confirmation, and six lack a way to revoke suspicious authorizations. We responsibly disclosed these issues to the affected vendors, several of whom acknowledged the problems and responded positively. We further conduct a user study with 100 participants, uncovering that the vast majority consider these rights essential and expect them to be upheld in *XDAuth*. Our study reveals a clear gap between current implementations and user expectations, underscoring the need for stronger user rights support to develop more secure, user-centered *XDAuth*.

I. Introduction

Currently, users frequently interact with online services across multiple devices, including phones, tablets, and personal computers. Correspondingly, the landscape of user authentication has evolved beyond single-device to a couple of devices. This paradigm shift has given rise to cross-device authentication (abbreviated as *XDAuth*), a critical mechanism that facilitates seamless and secure access to digital accounts across multiple devices. *XDAuth* refers to an authentication model in which a user initiates login on one device (the *target device*) and completes the authentication on another trusted device (the *authentication device*), as shown in Figure 1. The authentication device typically holds an active user session or securely stored credentials (e.g., passkeys), and is used to authorize the login without requiring the user to enter credentials on the target device.

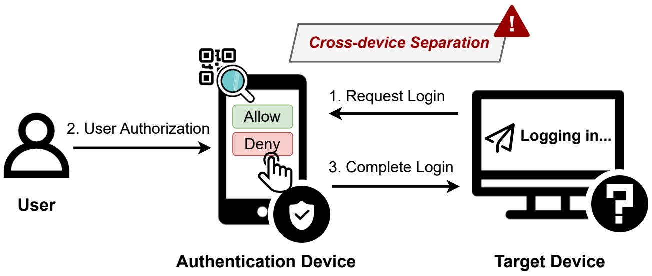


Fig. 1: Demonstration of the Cross-Device Authentication (*XDAuth*) mechanism.

In practice, the idea of *XDAuth* may be implemented through various schemes, where the most popular ones include QR code-based authentication, Push-based authentication, and WebAuthn [1]. For example, in a QR code-based authentication process, a target device displays a QR code and is then scanned by apps from the authentication device for users to authorize the login. In this way, users can simply scan a QR code and then click the “Allow” button (Figure 1) without entering a password. As a result, *XDAuth* can greatly simplify the login process and enhance user experiences.

However, the physical and contextual separation between the authentication and target devices introduces new security risks. This separation disrupts the expected contextual flow of authentication: the device initializing the request and the device authorizing it no longer share the same situational cues, creating information asymmetry. Such asymmetry can unintentionally lead to accidental deceptive patterns. These deceptions do not arise from malicious intent by service providers but from incomplete or ambiguous information presentation that impairs users’ ability to make fully informed

Network and Distributed System Security (NDSS) Symposium 2026
23-27 February 2026, San Diego, CA, USA
ISBN 979-8-9919276-8-0
https://dx.doi.org/10.14722/ndss.2026.240656
www.ndss-symposium.org

authentication decisions. In practice, diverse usage scenarios and the lack of standardized design principles further amplify these issues. Some implementations often omit key elements like contextual clarity, explicit user confirmation, or effective revocation mechanisms. As a result, users may lose track of which device is requesting authorization, mistakenly approve malicious login requests that exploit these gaps, and lack the ability to revoke unintended approvals. Such situations can lead to account compromise under modern threats such as social engineering [2], [3] and malware [4].

In this paper, to address these risks, we adopt a user-centric perspective and identify three essential user rights that should be supported throughout the *XDAuth* process: the right to know, the right to consent, and the right to control. Adapted from principles originally developed in the privacy domain, these rights are reframed as analytical lenses for ensuring clear, fully informed communication of authentication in contexts where information flows are fragmented across devices. These rights correspond to three key stages of *XDAuth* workflow. 1) In the *pre-authentication* stage, users should be fully informed of details about the target device and authorization activity (**right to know**). 2) During the *authentication* stage, explicit user consent is critical to prevent unintended access (**right to consent**). 3) In the *post-authentication* stage, users should be able to revoke any suspicious authorizations to terminate ongoing threats (**right to control**). Then we construct a tailored evaluation framework with concrete, measurable metrics to assess these rights across different stages. For example, to assess the *right to know* during the pre-authentication stage, we define six specific metrics that capture key aspects of the target device and the authorization activity, enabling users to make informed decisions.

Using our evaluation framework, we systematically assessed 27 major services that adopt three representative *XDAuth* mechanisms: QR code-based authentication, Push-based authentication, and WebAuthn. Our findings revealed that none of these services fully guarantees all three rights. Specifically, for the *right to know*, over half provide no information about the target device, like device types, for users to judge whether the request is legitimate. For the *right to consent*, some services implement ambiguous denial mechanisms and even allow the target device, which may be controlled by attackers, to determine the validity period of the authorization, introducing significant security risks. For the *right to control*, six services do not offer revocation capabilities after authentication. Even among those that do, some implementations are flawed. For example, a top short video platform with more than a billion downloads allows a revoked session to continue accessing and monitoring the user's chat history. We have reported these issues to the affected service providers, and as of the time of submission, we have received positive acknowledgments from *Zoho OneAuth* [5], *GitHub* [6], etc.

Furthermore, we conducted a user study with 100 participants to evaluate the perceived value of these rights from the users' perspective. The results showed that the majority consider these rights essential and expect them to be upheld in real-world *XDAuth* deployments. Besides, 98% believed these rights can enhance *XDAuth* security, and 95% considered the usability acceptable. These findings suggest that strengthening user rights in *XDAuth* not only aligns with user expectations but also enhances security without compromising usability.

In summary, we make the following contributions:

- We conduct the first empirical study on the usable security of *XDAuth*, focusing on the users' rights to know, consent, and control, to help better mitigate potential risks across different stages of the *XDAuth* workflow.
- We establish an evaluation framework with concrete metrics, and systematically assess 27 major services spanning three representative *XDAuth* mechanisms, revealing significant deficiencies in safeguarding these user rights. We responsibly report the issues to developers and receive acknowledgments from them.
- We conduct a user study[1] to evaluate users' perceptions on the user rights, finding that most users value these rights, expect them to be protected in *XDAuth*, and believe they won't impact usability.

II. Related Work

Usable Security of Authentication. Usability is a critical factor in the effectiveness of authentication systems, as users often avoid or misuse mechanisms they perceive as confusing or burdensome. Prior work has explored this issue from two main aspects: understanding user perceptions and improving system design. On the perception side, studies have shown that users often struggle with two-factor authentication (2FA) due to its complexity and device compatibility issues [7], [8]. Even with more modern methods like WebAuthn, usability remains a concern—users often misunderstand how their biometric data is handled, which undermines trust [9]. Similarly, vague or context-poor login alerts in risk-based authentication often leave users confused and unable to take correct actions [10]. These findings underscore the importance of clear communication and informed user decisions, which is one of the principles our work builds upon.

On the design side, studies have explored user interfaces and developer resources, such as identifying barriers to FIDO2 deployment and limitations in developer documentation [11], [12]. Some studies have assessed the consistency of 2FA workflows and found that the inconsistent design of 2FA user journeys leads to user confusion and cognitive overload [13]. Other work evaluated the effectiveness of suspicious login notifications and found that users preferred notifications that contain detailed information [14]. Overall, evaluations show that no single authentication method effectively balances usability, security, and deployability [15].

While these studies have deepened our understanding of how to make authentication more usable, they mainly focus on issues in single-device scenarios, whether single-factor or

[1] The raw data of our user study is available at https://github.com/XDAuth-security/XDAuth.

two-factor authentication. The growing use of cross-device authentication introduces unique interaction patterns and security challenges that remain largely unaddressed. Our work fills this gap by systematically studying the usable security of cross-device authentication and proposing user rights tailored to its distinct threats and interaction models.

Security and Usability of *XDAuth*. As authentication workflows increasingly span multiple devices, mechanisms like QR code login, push-based approval, and WebAuthn offer greater convenience but also raise new security and usability concerns. Existing studies mostly focus on issues of specific authentication schemes, identifying vulnerabilities in QR code-based login implementations [16] and usability barriers in WebAuthn [17], [18], [19]. Besides, the separation between devices in *XDAuth* opens the door to social engineering attacks, as shown in push-based authentication, where users can be tricked into approving unauthorized requests [20], [21]. These attacks highlight the need for better user safeguards. Our work aims to address this gap by proposing mitigations from the user's perspective that enhance decision-making and recovery in such scenarios.

While prior work has explored the usability of multi-device usage, it rarely focuses on the critical topic—authentication. Studies have examined fragmented workflows [22], cross-device privacy tracking risks [23], [24], and seamless session transfer [25], but overlook the security and usability challenges specific to authentication. Instead, our work fills this gap by focusing on how this device-separated authentication affects user interaction and trust, demanding more attention to awareness, consent, and control.

III. Problem Statement

A. Cross-Device Authentication

***XDAuth* Definition.** As illustrated in Figure 1, *XDAuth* refers to an authentication mechanism in which a user initiates login on one device and completes the authentication on another trusted device, which already holds an active session or securely stored credentials (e.g., passkeys). In this paper, we refer to the first device as the *authentication device* and the second device as the *target device*. This design allows users to approve login requests without entering credentials on the target device, enabling seamless and often passwordless access across multiple devices. A typical example is using a logged-in app on a smartphone to authorize login on the website accessed via a personal computer without entering credentials.

It is important to distinguish *XDAuth* from two-channel authentication methods, where users receive a one-time password (OTP) via SMS, email, or an authenticator app on a secondary device, and then manually enter it on the device initiating the login (the target device). In such cases, the secondary device serves only as a delivery channel for the OTP, while the login is finalized on the target device. In contrast, *XDAuth* involves active interaction solely on the authentication device, such as approving a prompt or scanning a QR code, without requiring the user to return to the target device to complete the login with credential input.

XDAuth can function as a standalone authentication method or be integrated into a multi-factor authentication (MFA) flow. Its core characteristic is that trust is transferred between devices through direct user action on the authentication device, thereby streamlining the login experience while minimizing credential exposure.

Analysis Scope. In this study, we focus on three widely adopted *XDAuth* methods: QR code-based authentication, push-based authentication, and WebAuthn [1], as summarized in Table I. These three mechanisms represent the most popular forms of *XDAuth* in current real-world deployments and serve as the scope of our analysis. We provide a brief overview of each in the following.

TABLE I: Three types of typical *XDAuth* mechanisms.

Type	Description	Examples
QR Code-based	Using the authentication device to scan a QR code shown on the target device.	WhatsApp, TikTok
Push-based	Receiving a notification on the authentication device to approve the login on the target device.	Facebook, Microsoft
WebAuthn	Using the authentication device to approve the login request based on cryptographic credentials.	Google, Apple

QR code-based authentication allows the user to scan a QR code displayed on the target device using a trusted authentication device, without requiring manual credential entry. It is commonly used in messaging and social media platforms such as *WhatsApp* and *TikTok*, where users frequently switch between mobile and desktop devices.

Push-based authentication sends a login notification to the authentication device, where the user can explicitly approve or reject the notification request, enabling authentication decisions. This mechanism is widely adopted by services like *Facebook* and *Microsoft*, especially in contexts requiring multi-factor authentication.

WebAuthn supports cross-device authentication by allowing users to authorize login requests on an authentication device based on public key cryptography [1]. Users first link their device by scanning a QR code shown by the target device. After this initial pairing, future logins can be requested via push notifications without rescanning. Unlike QR or push-based schemes, WebAuthn relies on cryptographic key pairs stored on the authentication device, enabling secure, passwordless authentication across platforms. It is now widely adopted by major providers such as *Google* and *Apple*.

We focus on mechanisms with interactive web or app-based interfaces, as they offer stronger potential to support user-facing operations. In contrast, we do not consider schemes like OAuth 2.0 Device Authorization Grant [26] and FIDO hardware security keys (e.g., Yubikey [27]), which are typically used on input-constrained or low-interaction devices, where support for active user rights is inherently limited.

These three *XDAuth* mechanisms reduce users' burden by replacing password entry while enhancing security. It has seen an increase in adoption across various service domains. However, the user-facing design of these systems raises important questions about whether users are adequately informed, empowered, and in control throughout the authentication process.

B. Motivation

This paper focuses on the usable security of *XDAuth*. While *XDAuth* improves usability by allowing users to authenticate across multiple devices, its distributed nature also introduces new risks, especially from social engineering and phishing. As illustrated in the IETF cross-device security draft [28], attackers can exploit the unauthenticated channel between the two devices and persuade users to grant authorization to gain unauthorized access. Unlike traditional single-device logins, *XDAuth* workflows often rely on user approval on a second device, which can be manipulated by attackers if users lack sufficient context or control.

A prominent motivating example is QRLJacking reported by OWASP [3], where attackers trick users into scanning a malicious QR code that mimics a legitimate login page. As shown in Figure 2, if users do not receive sufficient information about the target device and authorization activity, or if their consent is not explicitly required, they may unknowingly authorize a login session controlled by the attacker, leading to account takeover and privacy breaches.

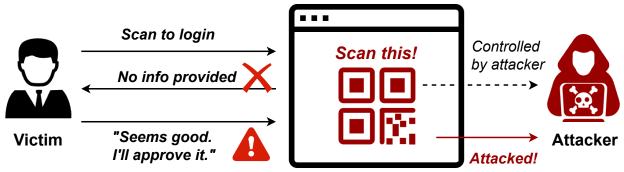


Fig. 2: The example of QRLJacking that demonstrates the importance of user rights.

These risks reveal a core challenge in *XDAuth* design: users are central to the authentication process, but lack sufficient support to make informed, secure decisions. Without clear information such as device identity, session details, or real-time feedback, and without the ability to reject or revoke access, users may be exposed to attacks. Thus, securing *XDAuth* requires more than technical protections. It demands a user-centric approach that enforces the rights to awareness, consent, and control throughout the entire login process. We argue that these rights are not optional usability enhancements but essential to secure and trustworthy *XDAuth* systems.

Threat Model. Our threat model focuses on *XDAuth*-specific risks from authentication-target device separation, which creates inherent information asymmetry: users approve requests on one device without directly observing the environment or state of the other. Adversaries can exploit this by initiating login requests from a device under their control (the malicious target) and deceiving users into approving them via their legitimate authentication devices. Unlike traditional credential phishing, which fabricates fake websites and steals passwords, this abuses legitimate *XDAuth* workflows and contextual confusion, not technical vulnerabilities. We exclude conventional password theft or network attacks, focusing on cross-device deceptive risks in *XDAuth*.

From the perspective of contextual integrity [29], these threats arise because *XDAuth* disrupts the expected flow of information within the authentication context. In traditional single-device authentication, information about the login action (e.g., purpose, device state, and environment) remains within a coherent context the user can interpret. Cross-device workflows, however, transfer authorization information across distinct contexts, breaking these situational boundaries. When contextual cues are fragmented, users can no longer verify authorization request alignment with their expectations, making them vulnerable to deceptive approvals. This work thus treats the restoration of contextual integrity through reinforcing awareness, consent, and control as key to mitigating such threats.

IV. *XDAuth* Analysis

To systematically assess the security and usability of *XDAuth* mechanisms, we begin by analyzing the general structure of the *XDAuth* workflow and identifying key user rights that should be preserved throughout the process.

A. Workflow of XDAuth

As shown in Figure 3, the workflow of *XDAuth* generally consists of three key stages: *pre-authentication*, *during-authentication*, and *post-authentication*. Each stage involves distinct user actions and system behaviors across the authentication and target devices. We provide a detailed discussion of each stage below.

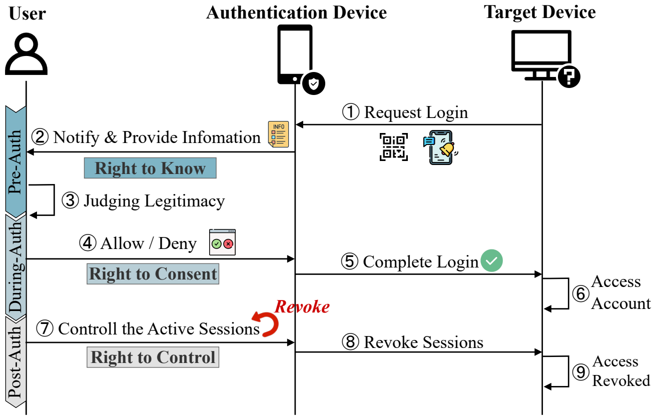


Fig. 3: The abstracted workflow of *XDAuth*.

Pre-authentication. This stage begins when the user initiates a login attempt on the target device. The system communicates with the authentication device using methods such as QR code scanning or push notifications (①). Then the user is notified on the authentication device and prompted to respond to the login request (②). Based on this information, the user evaluates the legitimacy of the request (③).

The goal of this stage is to provide users with sufficient contextual information to accurately identify the target device and the source of the login attempt. However, the physical and contextual separation between devices can limit the user's ability to verify whether the request is legitimate. Attackers may exploit this gap, for example, by replacing a QR code with a malicious one, tricking users into authorizing access to unintended devices.

During-authentication. In this stage, the user interacts with the authentication device to approve or deny the request (④). Since this device is already logged in and trusted, users are usually not required to re-authenticate. Instead, they are asked to confirm the action through simple interactions such as tapping a button or confirming biometric input.

The goal of this stage is to ensure that the user's decision is explicit and intentional. However, unclear interface designs or insufficient interaction requirements can cause users to approve requests unintentionally.

Post-authentication. Once the user approves the request, the target device is granted access (⑤) and begins operating under an authenticated session (⑥). At this point, the user may no longer have direct visibility into the session's status or behavior, especially when the device is temporary or shared.

The goal of this stage is to ensure users can monitor and manage ongoing sessions. Ideally, users should be able to control active sessions (⑦), revoke access to any suspicious or unrecognized session (⑧), and confirm that access has been successfully terminated (⑨). In practice, many systems lack mechanisms for post-login notifications or remote session termination, leaving users without effective means to respond to unauthorized access.

B. User Rights in XDAuth

Building on the goals of each stage in the *XDAuth* workflow, we propose three essential user rights—the *right to know*, *right to consent*, and *right to control*—that should guide the design of both secure and user-centered *XDAuth* systems. These rights are adapted from principles or originally developed in the privacy domain, but reframed here as analytical lenses for understanding the accidental deception and information asymmetry in authentication rather than ownership of data. This cross-domain adaptation reflects a paradigm-style exploration: applying privacy-derived models to usable security, where the goal is not to manage information disclosure but to ensure clear, fully informed communication of authentication.

These rights are grounded in established security and privacy principles from standards such as GDPR [30], CPRA [31], and the NIST frameworks [32], [33], which emphasize transparency, agency, and accountability. The IETF cross-device security draft [28] also highlights the risks introduced by unauthenticated cross-device channels and explicitly states that "**the only mitigation against this unauthenticated channel is the user's judgement**." In addition, guided by the analysis of autonomy harms [34] arising from failure to inform, manipulation, and lack of control, we frame these rights as safeguards that preserve users' ability to make informed, self-directed authentication decisions. Together, these standards and theoretical foundations support the three user rights that correspond to each stage of the *XDAuth* process, serving as actionable principles for assessing and improving real-world deployments in ways that strengthen both security and user autonomy.

Right to Know. In the pre-authentication stage, users often face difficulties in understanding the details of the authentication request and assessing the legitimacy of the requesting device. This lack of transparency can lead to uninformed decisions and increased security risks, such as granting access to malicious or unauthorized devices. The importance of transparency is well established in both regulation and research. For instance, GDPR [30] and CPRA [31] grant individuals the right to receive clear and relevant information before making decisions about their data. Similarly, prior work [21] highlights that missing contextual information in login prompts may compromise 2FA systems. In the context of *XDAuth*, the IETF cross-device security draft [28] also emphasizes the importance of "*providing better information with which to make decisions to authenticate the channel*" as a key mitigation against cross-device risks. By ensuring that sufficient context is available for users, the right to know mitigates the autonomy harm of failure to inform, enabling users to make informed choices with clear information.

Right to Consent. In the during-authentication stage, users must determine whether an authentication request is legitimate and then explicitly allow or deny it. However, the lack of mechanisms to ensure explicit consent from the user will increase the risk of accidental or unauthorized approvals. Regulatory principles such as GDPR [30] emphasize that consent must be freely given through a clear affirmative action, and prior work has similarly underscored this need. For example, Bonneau et al. [15] identified explicit consent as a critical security feature in authentication, arguing that the process should be actively triggered and explicitly approved by the user. Building on this, we propose the right to consent to foster user trust and improve security. This directly addresses autonomy harms stemming from manipulation, as insufficient consent flows can subtly influence users' decision-making.

Right to Control. In the post-authentication stage, users should be able to keep control over their active sessions to prevent prolonged or unnoticed misuse of access. Without such control, unintended or malicious authorizations may persist without the user's awareness. This is motivated by standards like NIST frameworks [32], [33], which emphasize the need for revocation when users detect compromise or no longer require access. The IETF cross-device security draft [28] also highlights the importance of "recovering from incorrect channel authentication decisions by users," reinforcing the need for post-login intervention. Building on these principles, we propose the right to control to ensure users can detect and respond to unauthorized or unwanted access, mitigating against autonomy harms associated with a lack of control.

V. Evaluating User Rights in *XDAuth*

Building on the user rights proposed for each stage of the *XDAuth* workflow, this section aims to identify specific aspects and metrics for effectively assessing each right. We first introduce our methodology to derive these metrics, which involves collecting real-world services, exercising and documenting their workflows, and coding the specific metrics. Based on this analysis, we construct an evaluation framework that maps each user right to a set of measurable criteria. As shown in Table II, these metrics allow us to systematically assess how well real-world services support user awareness, consent, and control in *XDAuth*.

A. Methodology for Deriving Evaluation Criteria

This subsection describes how we constructed the dataset and derived our evaluation framework. Our process includes three steps: 1) selecting major real-world *XDAuth* services, 2) documenting their *XDAuth* workflows, and 3) deriving evaluation metrics through coding-based analysis.

Dataset. We collected 27 widely used authentication services that implement one of three typical *XDAuth* schemes (QR code-based, push-based, or WebAuthn) [2]. To ensure diversity and representativeness, we selected 10 popular service categories (e.g., social networking, shopping, technology) from Sitereview [35]. For each category, we selected the top 10 websites using the Tranco top site list [36], resulting in 100 representative candidates. Then we manually tested each service for *XDAuth* support, identifying 23 services. Additionally, recognizing that authenticator applications often implement rich authentication flows, we further included four popular authenticator services, resulting in 27 services across 10 categories.

***XDAuth* Workflow Documentation.** Based on the collected 27 services, we systematically documented the *XDAuth* workflows of each service to identify specific metrics for evaluating user right support. Specifically, three researchers independently interacted with each service and documented its *XDAuth* workflow using notes and screenshots. Observations were guided by predefined goals aligned with the three user rights: 1) For *right to know*, we examined whether users receive sufficient contextual information about the authentication request and the environment of the target device. 2) For *right to consent*, we focused on whether decision points (e.g., approve/deny) and related authorization settings are clearly presented. 3) For *right to control*, we observed whether users receive feedback and have options to review and manage authorization events. We conducted two rounds of documentation. The first exploratory round captured all observed details related to the three rights, including any newly emerging aspects. We then consolidated these observations into a unified set of documentation dimensions and performed a second round to ensure consistent recording across all 27 services under the same dimensions.

Metrics Identification. Because no existing framework evaluates user rights in *XDAuth*, we applied grounded theory [37] to identify specific evaluation metrics. All login processes share a common framework of three user rights but differ in implementations, thus qualifying as semi-structured observational subjects. During open coding, three researchers extracted elements relevant to user rights from the workflow records and assigned conceptual labels. In axial coding, these concepts were grouped inductively, such as combining "IP address" and "geographical location" into "location & network information" to form broader, more generalizable categories. Coding was refined through multiple discussions to resolve disagreements, following recommended reliability practices [38].

Testing Procedure. All tests were approved by the Institutional Review Board (IRB) of our university and strictly adhere to established ethical standards in security research. We used researcher-owned accounts and devices, interacted only with publicly accessible interfaces, and avoided actions affecting system integrity or user data. Tests were primarily conducted on a Windows 11 desktop and a Pixel 8 (Android 15), with an iPhone 16 used where iOS was required.

B. Metrics for Right to Know

Building on our methodology, we identify concrete metrics for evaluating the rights as listed in Table II. The first one, the right to know, requires that users receive sufficient context about an authentication request. Our analysis yields two categories of information: authorization activity information and target device environment information.

Authorization Activity Information. This information helps users assess the legitimacy of an authentication request. We evaluate three specific elements:

- *Purpose of authorization.* This refers to whether the system clearly presents the purpose of the request (e.g., authorizing access to log in to the user's account). Clear descriptions receive ●; missing descriptions ○. This metric is critical because missing purpose information can increase users' cognitive ambiguity, making authorization requests from malicious attackers appear deceptive, thus directly affecting the perceived legitimacy.
- *Time of authorization.* This evaluates whether the time of the request is shown. Providing exact time earns ●; absence earns ○. In particular, *XDAuth* has an active pattern: users actively initiate authentication using the authentication device (e.g., scanning a QR code), where authorization time is inherently known. Thus, we mark it as ☒ in such cases. In the passive pattern, where the authentication device receives requests passively (e.g., push-based authentication), time information helps users correlate the authrozation with their expected requests to detect suspicious ones.
- *Granted capabilities and data.* This assesses whether users are informed about what capabilities and data are being granted. Systems are rated ● when stating this information and ○ otherwise. This information helps users understand authorization scope and potential consequences. Missing

[2] The complete list appears in Table III.

TABLE II: Three user rights and the evaluation framework with corresponding metrics.

Stage	Right	Category	Metrics	Description
Pre-Auth	Know	Authorization activity	Purpose	What this request is for.
			Time	When the login request is made.
			Capabilities & data	What capabilities and data the login request asks for.
		Target device environment	Device info	Details about the target device.
			Location & network	Where the target device is and what network it's using.
			Device risks	Whether the target device is trusted or risky.
During-Auth	Consent	Explicit authorization	Explicit consent & rejection	Let users clearly allow or deny the request.
		Duration	Agreement on duration	Whether the target device will be remembered for future logins.
Post-Auth	Control	Notification	Login notification	A message sent to users after a login happens.
		Authorization review	Ease of finding	Users can easily find and view their active login sessions.
			Session details	Users can see details of active sessions.
		Authorization revocation	Revocable session	Users can revoke any login they no longer trust.

descriptions may lead users to unknowingly grant excessive capabilities, increasing practical deceptiveness.

Target Device Environment Information. This information helps users confirm whether the requesting device matches their expectations and reduces information asymmetry caused by cross-device separation. There are three specific data points:

- *Device information.* Detailed device information helps users verify if the requesting device is their own or expected. As real-world information granularity varies and more precise details aid device recognition, we evaluate it by granularity: Specific models (e.g., "ThinkPad X1") as ●, general system information (e.g., Windows) as ◐, and coarser or missing identifiers (e.g., browser only) as ○. Granularity matters because generic identifiers often fail to distinguish legitimate devices from attackers' devices.
- *Location and network information.* Providing location and network data helps with legitimacy checks. Detailed geographic information earns ●; IP-only earns ◐ due to being unintuitive for regular users; absence earns ○. This acts as an intuitive anomaly signal.
- *Device risks.* This evaluates whether the system indicates unfamiliar or first-time devices. Presence earns ●; absence ○. The risk warning is encouraged and practical as even simple heuristics (e.g., first-time login on that device) effectively raise awareness and mitigate deception risks, and implementation is feasible given that most services collect device data.

To safeguard the right to know, *XDAuth* systems should present the above information clearly and accessibly on the authentication device. This enables informed decisions, reduces unauthorized access risks, and builds user trust.

C. Metrics for Right to Consent

The right to consent is evaluated on two aspects: explicit authorization and duration agreement, capturing how user intent is respected and deceptive patterns are prevented.

Explicit Consent and Rejection. This evaluates whether systems provide balanced and explicit approval and rejection mechanisms. Systems offering both clear options are rated ●; approval-only interfaces are ◐; bypassing approval is ○. This metric is essential because lacking an explicit consent or highlighting only approval without rejection may prompt users into unintended consent.

Agreement on Authorization Duration. Users should be able to choose whether access is temporary or remembered. Providing explicit choices (e.g., "this session only", "remember this device") on the authentication device earns ●. Default temporary authorization without explicit choice is ◐. Systems that default to persistent authorization or provide options only on the target device are rated ○, as these patterns may allow prolonged unauthorized sessions after compromise.

To uphold the right to consent, *XDAuth* systems should use clear, balanced interfaces with equally prominent approval and rejection options, and let users choose authorization duration to prevent persistent malicious access. Explicit, user-driven consent strengthens security against unauthorized access and reinforces trust by respecting user intent and control.

D. Metrics for Right to Control

We evaluate real-world support for the right to control across three progressive capabilities: notification, authorization review, and authorization revocation.

Notification. This assesses whether users are promptly informed when new authorization sessions occur, a key first step for ongoing control. We classify implementations into three levels by notification channel and visibility. Persistent alerts (e.g., in-app messages, email, SMS) earn ●; temporary notifications (e.g., pop-ups, transitional pages) earn ◐ as they are easily missed unless users are highly attentive; no notifications earn ○. Timely and persistent notifications are crucial because unnoticed login events may lead to deceptive or prolonged compromise in *XDAuth*, and invisible alerts leave users unaware of unauthorized sessions.

Authorization Review. This aspect is critical for users to monitor authorization activities and verify legitimacy, staying informed after login. We assess it using two factors:

- *Ease of finding.* How easily users can find the interfaces displaying active authorizations is essential for effective review and control. We assess it using "page path length", defined as how many clicks or steps are needed from the homepage. A single step is ideal, though practical designs often need extra steps for function organization. Nevertheless, shorter paths make the feature easier to notice and use, while deeply buried functionality reduces its visibility and makes users less likely to review or manage ongoing authorization requests.

- *Session details.* Systems should provide existing authorization session details for review. These largely overlap with right-to-know metrics, including authorization time, device, and location, plus login method (e.g., QR code, password) to help users verify alignment with their expectations. Shared criteria match the right to know; login method visibility earns ● if provided, ○ if absent. In this functionality, single cues (e.g., location) are ambiguous as attackers may share attributes like city. Thus, combining multiple contextual factors improves differentiation, aligning with prior findings [39], [10] that richer information enhances suspicious or unauthorized login detection.

Authorization Revocation. Users should be able to promptly revoke uunauthorized or malicious sessions after reviewing existing ones to mitigate persistent threats. We evaluate this capability by the revocation feature availability and usability. Direct, usable revocation earns ●; revocation provided but compromising usability earns ◐; lack of revocation option earns ○. Authorization revocation enables post-authentication control, and its absence turns one-time mistakes into sustained exploitation.

Overall, these metrics characterize how *XDAuth* empowers users to maintain ongoing awareness and control. Notifications and review enable real-time monitoring and anomaly detection. Revocation supports prompt risk mitigation, empowering proactive user action, reducing prolonged unauthorized access risks, and enhancing *XDAuth* security.

VI. Evaluation Results

Based on the evaluation framework, three researchers independently assessed real-world *XDAuth* deployments. Fleiss' Kappa (κ) for 15 metrics showed high inter-rater reliability: a mean $\kappa = 0.982$ (near-perfect), with substantial agreement for "Granted capabilities and data" ($\kappa = 0.844$) and "Authorization revocation" ($\kappa = 0.879$) and perfect agreement for the rest. All findings were cross-validated for consistency. Below we present evaluation results, identifying common deficiencies in supporting user rights and key improvement directions.

A. Overall Results

As shown in Table III, none of the 27 evaluated services fully safeguard all three user rights in *XDAuth*. While some services demonstrate partial support across different dimensions, no single implementation achieves comprehensive protection. For example, *Yandex* and *Uber* offer relatively strong coverage across multiple rights, but still lack in areas such as authorization duration control. In contrast, services like *Apple* and *Xero Verify* perform poorly across all three rights.

Among the three rights, the right to consent is the most consistently enforced. Nearly all services implement some form of explicit user approval through button taps or biometric confirmation before granting access. However, some enforcements offer no management over session duration. The right to control is less supported: nearly a quarter of the services fail to provide revocation options, while notification of login activity and accessible session review features are frequently missing or poorly integrated.

The right to know is often only partially supported. Although some services display basic request information, few provide sufficient details about the target device or the environment, such as geolocation, device types, or risk alerts. This lack of contextual information limits users' ability to evaluate the legitimacy of authentication requests, increasing the risk of social engineering attacks or unintentional approvals.

Additionally, we observed that WebAuthn implementations by *Apple* and *Google* use Bluetooth Low Energy (BLE) to enforce proximity during authentication, which can help reduce certain threats. However, as noted in the IETF cross-device security draft [28], WebAuthn may implement these user rights to further strengthen security. QR code-based and push-based schemes lack such distance constraints and therefore should ensure strong support for user-facing protections.

Key Insight 1 (overall results): Despite the critical role of users in *XDAuth*, none of the evaluated services fully safeguard all three user rights. The right to consent is most widely supported, while the right to know is often poorly implemented, especially in conveying device and environment details. The right to control shows mixed implementation quality. Overall, support across rights is uneven, leaving significant room for improvement.

B. Detailed Analysis of User Rights

Right to Know. The right to know enables users to assess whether an *XDAuth* request is legitimate by providing clear, contextual information. However, our evaluation shows that this right is often insufficiently supported. Among the 27 services, over half (14) provide no device or environment information (e.g., model, geolocation), leaving users without cues to verify the authenticity of the target device. For instance, *TikTok* displays only a generic message ("Confirm login to TikTok on the PC") without specifying any device information. Additionally, only three services explicitly disclose capability scopes (e.g., access to contacts, managing private accounts), while others entirely obscure such details. Moreover, only *QQ* alerts users to environmental risks (e.g., suspicious login), making it harder to identify potential threats.

Even when information is provided, it is often incomplete or presented in unhelpful formats. For example, *Keeper* displays IP addresses, which in theory could help infer the login location. However, non-technical users may find it challenging to correlate these IP addresses with specific devices without direct geographical information. Some other services disclose only coarse-grained location data, such as country-level ge-

TABLE III: Evaluating results for user rights in 27 real-world *XDAuth* services.

Mechanisms	Service	Category	Right to Know		Right to Consent		Right to Control		
			Authorization Activity Info[1] $(Pur,Tm,Capability)$	Device and Environment Info[2] $(Dev,Loc,Risk)$	Explicit Authorization	Agreement on Durations	Notification	Authorization Review[3] $(Pth/Tm,Mtd,Dev,Loc)$	Authorization Revocation
QR Code-based Authentication	WhatsApp	Chat (IM)/SMS	● ☒[†]●	○ ○ ○	☒	○	○	2 / ● ○ ◐ ○	●
	Telegram	Chat (IM)/SMS	○ ☒ ○	○ ○ ○	☒	◐	◐	3 / ● ○ ● ●	●
	TikTok	Audio/Video Clips	● ☒ ●	○ ○ ○	◐	◐	○	5 / ● ● ● ○	●
	Steam	Games	● ☒ ○	○ ● ○	●	●	◐	3 / ● ● ◐ ●	●
	Yandex	Search Engines/Portals	● ☒ ○	◐ ● ○	◐	◐	○	x[‡]/ ○ ○ ○ ○	○
	Uber	Travel	● ☒ ○	◐ ● ○	●	◐	◐	x / ○ ○ ○ ○	○
	QQ	Chat (IM)/SMS	● ☒ ○	○ ● ●	●	◐	◐	4 / ● ○ ● ●	●
	Mail App	Search Engines/Portals	● ☒ ○	○ ○ ○	●	◐	◐	3 / ● ○ ◐ ●	●
	Roblox	Games	● ☒ ●	○ ● ○	●	◐	◐	3 / ● ○ ◐ ●	●
	Discord	Chat (IM)/SMS	● ☒ ○	○ ○ ○	●	◐	◐	3 / ● ○ ◐ ●	●
	Taobao	Shopping	● ☒ ○	○ ○ ○	●	○	○	4 / ● ● ● ●	●
	Baidu	Search Engines/Portals	● ☒ ○	○ ○ ○	●	◐	○	5 / ● ● ◐ ●	●
	VK	Social Networking	● ☒ ○	◐ ● ○	●	◐	●	4 / ● ○ ● ●	●
	Ivi	Entertainment	○ ☒ ○	○ ○ ○	○	◐	◐	2 / ○ ○ ● ○	●
	Weibo	Social Networking	● ☒ ○	○ ○ ○	●	◐	○	4 / ● ○ ● ●	●
Push-based Authentication	Facebook	Social Networking	● ● ○	◐ ● ○	●	◐	●	6 / ● ○ ● ●	●
	Microsoft	Technology/Internet	● ○ ○	◐ ● ○	●	○	◐	2 / ● ○ ◐ ●	◐
	Steam	Games	● ○ ○	○ ● ○	●	●	◐	3 / ● ● ◐ ●	●
	GitHub	Technology/Internet	● ○ ○	○ ○ ○	●	◐	◐	x / ○ ○ ○ ○	○
	Keeper	Technology/Internet	● ● ○	○ ◐ ○	●	◐	◐	3 / ● ○ ○ ●	○
	Google Prompt	Search Engines/Portals	● ● ○	◐ ● ○	●	○	●	4 / ● ○ ● ●	●
	Zoho OneAuth	Business/Economy	● ○ ○	○ ○ ○	●	◐	●	1 / ● ○ ● ●	●
	Xero Verify	Business/Economy	● ○ ○	○ ○ ○	●	○	○	x / ○ ○ ○ ○	○
	Wise	Business/Economy	● ○ ○	◐ ● ○	●	◐	○	x / ○ ○ ○ ○	●
	Snapchat	Chat (IM)/SMS	● ● ○	◐ ● ○	●	◐	◐	3 / ● ○ ◐ ●	●
WebAuthn	Apple	Technology/Internet	● ☒ ○	○ ○ ○	◐	◐	○	2 / ○ ○ ○ ○	○
	Google	Search Engines/Portals	● ☒ ○	○ ○ ○	●	●	○	4 / ● ○ ● ●	●

[†] "☒" indicates not applicable for this service.

[‡] "x" indicates that the service lacks a session review function.

[1] Authorization activity information includes, in order, purpose, time, and granted capabilities & data.

[2] Device and environment information refers to, in order, device information, location & network, and device risk.

[3] Authorization review includes, in order, page path length, login time, login method, device information, and location & network.

olocation (e.g., "United States"), which fails to highlight suspicious intra-country logins.

These issues reflect deeper challenges in *XDAuth* design. First, an overemphasis on usability creates a "security illusion", where over-simplified interfaces obscure critical details. While clean UI design improves user experience, it can also encourage users to approve vague or deceptive prompts (e.g., "Click Agree to Use the Service") without sufficient scrutiny. Second, a lack of industry standards often shifts responsibility to backend systems, assuming that security decisions should be automated rather than user-driven. In reality, the provider's context might not fully overlap with the user's context, and users are often well-positioned to identify unusual activity, particularly when they know their own trusted devices, if given adequate information.

Finally, the right to know raises a fundamental tension between privacy and transparency. Richer information aids risk detection but may reveal sensitive details, especially in shared-device scenarios. Future designs could address this trade-off through adaptive disclosure mechanisms, which present more detailed information only when risk levels are high or user context demands greater visibility, striking a balance between privacy protection and informed user decision-making.

Key Insight 2 (right to know): The right to know is often partially supported, with many services failing to present device or environment details. Future designs should move beyond generic prompts and adopt context-sensitive disclosures that balance transparency with privacy.

Right to Consent. The right to consent empowers users to make deliberate decisions during *XDAuth*, requiring explicit actions to decide authorization scope and duration. While most services enforce explicit consent (e.g., via button clicks), several design and implementation flaws weaken the effectiveness of this right.

First, a key issue observed is that some services still lack an explicit user consent step during authentication. For example, *Ivi*'s QR code-based login grants access immediately after scanning the QR code, without requiring user confirmation on the authentication device. This design can expose users to potential risks of social engineering attacks such as QRLJacking [3], where malicious QR codes of attackers are used to trick users into unintentionally authorizing access.

Even among services that enforce explicit consent, usability and security gaps remain. Four services lack a clear "Deny" button, forcing users to close the interface to refuse a login attempt, which can create confusion and hesitation. In some cases, visual design choices further weaken the rejection path, such as rendering “Deny” in low-contrast colors or placing it in a less prominent position than “Approve”. These patterns may discourage refusal and risk inadvertent approvals, manipulating users and causing autonomy harm.

Notably, *WhatsApp* and *Telegram* fall under the “not applicable” category for this criterion. In these two cases, the QR code scanner is only accessible after the user explicitly initiates the device linking process by tapping a “Link New Device” button, which already implies user consent. As a result, despite no separate “approve” step after scanning, the user’s prior action serves as a form of confirmation. Nevertheless, these variations in confirmation flows across services may lead to user confusion and weaken their mental models of secure login behavior, as also noted in prior work [13].

On the other hand, most services do not allow users to control the duration of authorization. Only three services offer options like “remember this device” on the authentication device, while the others do not provide users the right to decide this key authorization setting. Worse yet, some services (e.g., *WhatsApp*, *Microsoft*, *Google Prompt*) allow session duration decisions to be made on the target device. This violates the least-privilege principle, enabling attackers to extend access without user knowledge or consent once compromised.

These issues reveal that current *XDAuth* designs often rely on flawed assumptions about user intent. Treating actions like QR code scanning as implicit consent overlooks the risk of unintentional triggers or social engineering. The lack or poor visibility of denial options suggests a trade-off favoring convenience over control, where usability for seamless access comes at the cost of security clarity. Moreover, delegating critical decisions such as session duration to the target device, which is potentially attacker-controlled, reflects a systemic asymmetry in design responsibility, where control over risk is shifted away from the user. To truly uphold the right to consent, authentication systems should provide explicit confirmation, accessible refusal, and user-controlled session settings—all handled on the authentication device.

Key Insight 3 (right to consent): While most services support explicit consent, some skip it or use specially designed pages to encourage users toward approval, raising potential risks. Future systems should ensure consent is intentional and user-driven, with explicit prompts and full user control over key decisions.

Right to Control. The right to control is vital for keeping security after authentication, enabling users to monitor active sessions and revoke access when needed. However, our evaluation reveals that this right is significantly undermined by the lack of timely notification and limited session visibility.

First, many evaluated services (10/27) fail to provide any form of real-time notification when a login occurs, making it difficult for users to detect unauthorized access promptly, especially in designs that skip explicit consent. Even among those that do provide alerts, most (13/17) rely on temporary UI elements like brief toast messages that last only a second, making them easy to miss. Also, when users attempt to verify active sessions afterward, they often face further obstacles. Specifically, 5 services do not offer a session review function at all, while others may bury it deep within settings, significantly reducing its usability. For instance, *Facebook* requires digging through six menus just to access the review function.

Furthermore, the session management functions in most services are insufficient for users to take effective action. Six evaluated services provide no means to revoke active sessions, while for those that support revocation, the mechanisms may be unreliable or difficult to use. For example, *Microsoft* relies on password resets to revoke access to unauthorized devices, which imposes significant usability burdens and provides unclear feedback about whether the action is effective. Worse still, password changes do not terminate existing sessions, exposing flaws in session management. In a more serious case, a leading short video service fails to fully revoke access, allowing the revoked session to continue accessing the user’s real-time chat history.

These issues reflect a fundamental weakness in *XDAuth* systems design: the neglect of post-authentication security. To streamline the login experience, many services sacrifice visibility into security risks and reduce the ability to intervene after authorization. Features like silent notifications, hidden session review interfaces, and ineffective revocation mechanisms make it difficult for users to identify or respond to potential threats. This problem is further amplified in decentralized architectures. For example, *Google*’s password manager supports storing passkeys for multiple third-party services to achieve WebAuthn logins. While this improves usability, it fragments session control, as users cannot directly review or revoke all granted authorizations from a centralized place. Moving forward, future designs should better balance usability with user control, ensuring visibility, consistency, and revocation ability in the post-authentication stage so users can meaningfully retain full control over their active sessions.

Key Insight 4 (right to control): Current implementations of the right to control are fragmented, where many services lack timely notification, accessible session views, or effective revocation, leaving users unable to manage access. Future designs should ensure unified, transparent, and actionable post-authentication control.

Developer Feedback. We responsibly reported our evaluation findings and user rights recommendations to the service providers involved. As of the time of submission, several developers have responded and acknowledged our reports. For instance, the popular authenticator service *Zoho OneAuth* responded positively that our suggested feature has been added to their product roadmap and is currently under development. These interactions suggest a growing awareness among developers, but also highlight the need for more accessible and usable implementations of user-centric features.

VII. User Perceptions

Our evaluation of major services reveals critical gaps in how *XDAuth* implementations protect user rights, yet understanding user perspectives is equally essential to assess the practical importance and usability impact of these rights. We therefore conducted a user study to explore how real users perceive and value these rights in the *XDAuth* process. This section first presents the questionnaire-based user study's design and methodology, then presents its results and insights.

A. Design of the User Study

To better understand users' perspectives on *XDAuth* and their attitudes toward these three user rights, we conducted a user study guided by the following research questions:

- **RQ1**: How do users engage with *XDAuth*?
- **RQ2**: How do users perceive these three user rights?

To answer these questions, we designed an online questionnaire including 14 questions. The survey consisted of three main parts and was carefully crafted to ensure clarity and realism for general users. We began by introducing the concept of *XDAuth* using three illustrative diagrams based on real-world services. These help participants intuitively understand what *XDAuth* means. Participants who have never used such authentication methods were not included in the study.

The first part of the questionnaire focused on participants' use of *XDAuth*, including the specific schemes they adopt and how frequently they interact with them in daily life. The second part investigated user attitudes toward the three user rights. To help participants understand how these rights manifest during the *XDAuth* flow, we provided a demo video of a typical QR code–based login. For each right, we prepared two diagrams—one illustrating the absence of the right and one showing its presence—based on modified screenshots from real services with company names removed to avoid brand-related bias. Participants indicated their preferred version and explained their reasoning. Furthermore, we asked them about the impact of these user rights on security and usability. The final section collected demographic information for statistical analysis. The full questionnaire is provided in Appendix A.

Recruitment and Demographics. We recruited N = 100 participants from the United States via Prolific, a widely used online survey platform in usability and security studies [10], [13], [14]. The user study was conducted in March and April 2025. To ensure participants have *XDAuth* experience, we included a screening question following the guidelines of Prolific [40]. Each participant could submit only once. On average, participants took 8.5 minutes to complete the survey and received a compensation of $1.57, aligned with Prolific's recommended hourly rate.

Among the participants, 57% were female. The largest age groups were 18–30 (36%) and 31–45 (37%). Most participants (82%) held a college degree or higher, and 55% had a STEM background. Table V presents the demographic statistics.

Qualitative Analysis. For the three open-ended questions (Q4, Q6, Q8), we collected a total of 300 responses and conducted a qualitative code analysis. Two researchers first independently developed preliminary codebooks for each question using inductive analysis [41]. They then coded all responses separately to ensure comprehensive data coverage. Inter-coder reliability was assessed using Cohen's Kappa, with all categories achieving satisfactory values above 0.70 (mean Kappa value = 0.90) after iterative discussion and refinement. After resolving discrepancies, we finalized three codebooks (one per question) and performed frequency analysis (Appendix B-B).

B. User Perception Analysis

Through analyzing responses from 100 participants, we proceed to answer our research questions with the according findings. Overall, the results indicate that as a widely adopted mechanism, there is still room for *XDAuth* to improve. Users clearly value the rights to know, consent, and control, and prefer interfaces that incorporate these rights. Encouragingly, the integration of such rights does not appear to harm usability while improving security, suggesting the potential for practical deployment in real-world *XDAuth* systems.

RQ1: User Engagement. Among users with *XDAuth* experience, push-based and QR code–based authentication were the most widely used schemes, with nearly 80% having used one of them (Q1) while WebAuthn was slightly less commonly encountered. Over 60% reported using *XDAuth* daily or several times per week (Q2), indicating that it has become a regular part of their digital routines. This widespread and frequent use underscores the need for secure and user-respecting authentication experiences.

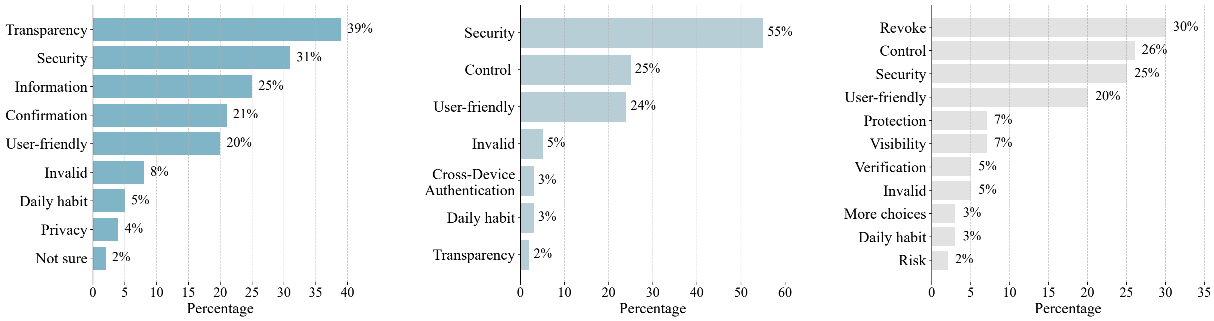
> **Key Insight 5 (*XDAuth* importance)**: *XDAuth* is integral to users' digital activities, and its high usage frequency underscores the need to ensure its security and usability.

RQ2: Perceptions of User Rights. When shown contrasting designs—one supporting a user right and one lacking it—participants overwhelmingly preferred the rights-preserving version (Table IV). These preferences were statistically robust (all $p < .001$) with large effect sizes, reflected by Risk Differences ranging from +46% to +82%. This demonstrates a strong user demand for more informed, intentional, and controllable authentication workflows.

TABLE IV: Statistical analysis of user preference between designs with and without user rights.

Question	Right	Design w/ Right	Design w/o Right	p-val	Risk Difference	95% CI for w/ Right
Q3	Know	91%	9%	$p < .001$	+82%	[+83.8%,+95.2%]
Q5	Consent	73%	27%	$p < .001$	+46%	[+63.6%,+80.7%]
Q7	Control	85%	15%	$p < .001$	+70%	[+76.7%,+90.7%]

We also examined whether user characteristics (age, education, professional background, usage frequency) influenced these preferences. Chi-square tests showed no significant associations for any factor (all $p > .05$; see Table VIII). Although not statistically significant, several descriptive patterns across



(a) Q4. Right to Know.

(b) Q6. Right to Consent.

(c) Q8. Right to Control.

Fig. 4: Response distributions for open-ended questions on the reasons behind user rights design choices.

different user subgroups were observed. Older participants (46+) expressed a stronger preference for the right to consent (81%) compared to younger users (70%). Non-STEM participants tended to favor explicit consent more than STEM users (82% vs. 65%). For the right to control, preferences were slightly higher among high-frequency users (88%) than low-frequency users (81%).

Notably, **98%** of participants believed that protecting these rights enhanced the overall security of *XDAuth* (Q9). We further analyzed participants' reasons behind their choices, as illustrated in Figure 4, to better understand the motivation driving their preferences.

Right to Know. In our survey, 91% of participants chose the design that explicitly shows the details of the login request and device (Q3). The primary reasons (70/91) emphasized transparency and security (Q4). For instance, one participant stated, *"I chose B because it offers transparency, helping users make informed decisions and detect suspicious devices or activities early."* Another user mentioned, *"It's safer in my opinion and the fact that we can figure out if it's someone that I know who is trying to log in."* This indicates that most users consider clear and detailed login request information as an essential factor for authentication decisions.

Among the 9 participants who did not choose the design supporting this right (Q3), 5 appeared to reflect confusion or misunderstanding (Q4)—for example, some mistakenly thought personal data would be displayed to the target device when requesting login, or selected the option conflicting with their stated reason. The other 4 cited reasons include preferring simplicity, following familiar patterns from daily use, or having privacy concerns. However, our usability measure shows that the overall usability of these rights is acceptable (Key Insight 6). Moreover, incorporating user rights protections barely changes familiar usage. For privacy, future designs could explore dynamic disclosure mechanisms (e.g., risk-based detailed information) to enhance protection.

Right to Consent. Most participants (73%) preferred the design explicitly supporting the right to consent in *XDAuth* (Q5). These users emphasized that requiring an explicit approval step enhanced security and control (Q6), preventing unauthorized access, especially in cases where the QR code is replaced by a malicious one. As one participant put it, *"I chose this option because it enhances security by giving users control over login approvals, preventing unauthorized access from unknown devices"*, while others highlighted that it *"lessens the opportunity for fraud"* and *"feels like I have more control."* Users also felt more empowered when given the ability to approve or reject access themselves, reinforcing a sense of control and reducing unintended authorization.

By contrast, 23 participants chose the design without explicit consent (Q5), primarily citing ease of use. This preference for simplicity over security is concerning. Besides, some believed scanning a QR code implied consent, while others were influenced by familiar designs. These responses reflect a misunderstanding of meaningful consent. Simply scanning does not ensure users fully grasp what they are authorizing or prevent them from scanning malicious codes and triggering an unauthorized login. Simplicity matters, but not at the cost of user control and protection. This calls for future research on balancing security and usability in *XDAuth* design.

Right to Control. 85% of participants chose the design supporting the right to control (Q7), indicating strong demand for post-authentication control mechanisms in *XDAuth*. Most highlighted that being able to view and log out active sessions or connected devices boosted security and personal control (Q8). This feature is seen as essential for mitigating risks like forgotten logins on shared devices, unauthorized access, or fraudulent activity. As one user explained, *"I chose this option because it provides greater control, enabling users to log out sessions or devices, improving security and privacy"*, while another noted, *"It empowers me to take action against fraud opportunists"*. Others appreciated the ease of managing sessions and described the design as more user-friendly. These responses reflect a common expectation of retaining post-login session review and revocation ability.

Only 15 participants opted against the design supporting the right to control (Q7), mainly prioritizing simplicity. However, such control features barely impact usability as they do not interfere with user experience when unused. One participant raised concern about others controlling their account, likely a misunderstanding, as the feature is designed to give control solely to the account owner. While simplicity is important, reducing user control may lead to greater security risks and a lack of control over ongoing sessions.

C. Interactive Validation Study

Considering the 100-participant survey relied on static screenshots and video demonstrations, which may not reflect the real implications of *XDAuth*. To strengthen the validity of our findings, we conducted an additional small-scale interactive study by implementing a functional *XDAuth* prototype. This follow-up study aimed to enable real user interactions with live systems to verify if the earlier preferences and trade-offs hold in practical contexts.

Study Design. The demo system we developed included a website to log in and a corresponding mobile app in an authenticated state. Participants were asked to log in to the website by scanning the QR code with the app, experiencing a simulated typical *XDAuth* workflow with a step-by-step interaction guide. To protect participants' privacy, the system used predefined simulated data (e.g., device information, location) and a demo account that avoided collecting real credentials or personal information. After completing the task, participants filled out a questionnaire used in the 100-participant survey to assess their perception of user rights. The questionnaire excluded the demo-video part, and the questions about participants' prior use of the three *XDAuth* schemes. Instead, after completing the interactive task, participants answered only the questions concerning the three user rights.

Recruitment and Findings. We recruited 10 U.S. participants via Prolific, following the same procedure as the main study. The participants interacted with the live system and completed the questionnaire in an average of 17.8 minutes, receiving \$3.32 each. The demographic statistics are presented in Table V. Results aligned with the earlier findings: participants strongly valued these three user rights in *XDAuth*. Eight preferred the design implementing the right to consent, nine favored the right to know and right to control. Moreover, eight considered these designs improved security without reducing usability. These consistent outcomes confirm that users' positive attitudes toward the rights persist in real interactions, supporting the robustness of our earlier conclusions.

Key Insight 6 (user perception on rights): Most users view the three user rights essential for *XDAuth* and expect them to be respected. 98% of participants believe these rights make *XDAuth* more secure, while 95% find usability acceptable.

VIII. Discussion

Implementation Recommendations for Developers. Drawing on our empirical evaluation and user study, we provide practical recommendations to enhance user protection in *XDAuth* systems. Implementing these features in practice requires a careful balance between security and usability.

Provide clarity without overload. Authorization interfaces should include key contextual cues (e.g., device information, location, request purpose) while avoiding cognitive overload. A progressive disclosure strategy can surface critical details first, with optional access to more detailed fields (e.g., folded until user expansion). Highlight anomalous attributes (e.g., unfamiliar device or location) from user patterns to help users quickly identify suspicious activity.

Maintain intentionality with minimal friction. Design consent mechanisms for deliberate user decisions without sacrificing efficiency. Use balanced approve and deny options, lightweight confirmations, and clear interaction hints to enforce explicit consent while streamlining flow. Ease of use should not be interpreted as cutting user actions at the expense of intentional choice, but minimizing unnecessary friction.

Offer reliable revocation with easy access. Session visibility and revocation options should be easy to find, not buried in deep settings. A centralized session overview with direct revocation offers strong corrective control when pre-authentication judgments fail. Notifications should appear when meaningful session changes occur and link directly to relevant controls.

Furthermore, the community should work toward developing a unified guideline or standard for upholding user rights, ensuring the security and usability of *XDAuth*. Major Internet service providers, with their significant influence, are well-positioned to lead this effort by setting examples and promoting best practices.

Comparison with Single-Device Authentication. Compared to traditional single-device authentication, *XDAuth* introduces both convenience and risks. Single-device logins maintain contextual integrity [29]: the same device initiates and confirms authentication, preserving immediate situational awareness. In contrast, cross-device workflows distribute information and control across devices, widening the context gap between user perception and target device state. This shift breaks the contextual integrity [29] by introducing information asymmetry, expanding the attack surface through unauthenticated cross-device channels. Therefore, secure and usable *XDAuth* design should strive to restore informational symmetry and user agency through transparent authorization cues, explicit consent mechanisms, and post-login control options, thus preserving informed, secure communication of authentication.

Limitation & Future Work. This study systematically analyzes how to make a clear, fully informed authentication communication in *XDAuth*, yet certain limitations remain. First, while our sample includes 27 widely-used services across diverse categories, it may not capture less common cases. Nevertheless, the rights identified are grounded in recurring patterns across services, suggesting that our findings are broadly indicative. Second, as with prior work [14], [10], our user study involved primarily U.S.-based participants. Cross-cultural studies could further validate the generalizability of awareness, consent, and control as core user rights. Third, while the 100-participant study used video and screenshots, we conducted an additional small-scale interactive study to validate the reliability of these findings under real *XDAuth* scenarios. As an online survey, users' potential self-report bias was mitigated via neutral wording and quality checks.

Future work could further explore translating high-level user rights into actionable, context/platform-specific design patterns and interface elements. Standardizing rights-centric practices may improve consistency and usability. Additionally, engaging developers directly through interviews or targeted surveys could also reveal implementation barriers and inform practical design guidance.

IX. Conclusion

This study identifies critical gaps in current *XDAuth* implementations to ensure clear, informed authentication communication. A systematic evaluation of 27 major services and a 100-participant user study show many systems lack sufficient transparency, explicit consent, and effective post-authentication control, which are capabilities users consistently value as essential. These gaps erode trust and expose users to threats like social engineering, session abuse, and unauthorized access. To mitigate risks, we advocate integrating the rights to know, consent, and control as core *XDAuth* design principles, calling for a shift to more user-centric, accountable mechanisms aligned with security standards and user expectations.

Acknowledgment

We thank the anonymous reviewers for their valuable feedback and constructive suggestions, which greatly improved the quality of this paper. Xiaohan Zhang is the corresponding author, and this work was supported in part by the National Natural Science Foundation of China (62102091, 62172104, 62172105, 62472096, 62102093, 62302101, 62402114, 62402116, 62202106), National Key Research and Development Program (2021YFB3101200).

Ethics Considerations

Both the main user study (N = 100 participants) and the follow-up interactive validation user study (N = 10 participants) were approved by the Institutional Review Board (IRB) of our university. All procedures followed established ethical standards for human-subject research in security and privacy. Participants in both studies were fully informed of the study purpose, procedures, and data practices before giving consent, and participation was voluntary with the option to withdraw at any time without consequence. Only non-sensitive demographic data were collected, all data were anonymized and securely stored, and no personally identifiable information was collected. In the follow-up validation study, participants were explicitly informed that all account and contextual device data were simulated to avoid collecting real user information.

References

[1] "Web authentication: An api for accessing public key credentials." https://www.w3.org/TR/webauthn/, 2021.

[2] K. Krombholz, H. Hobel, M. Huber, and E. Weippl, "Advanced social engineering attacks," *Journal of Information Security and applications*, vol. 22, pp. 113–122, 2015.

[3] "Qrljacking, an attack introduced on owasp." https://owasp.org/www-community/attacks/Qrljacking, 2024.

[4] H. Xu, M. Yao, R. Zhang, M. M. Dawoud, J. Park, and B. Saltaformaggio, "{DVa}: Extracting victims and abuse vectors from android accessibility malware," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 701–718.

[5] "Zoho oneauth." https://www.zoho.com/accounts/oneauth/, 2025.

[6] "Github." https://github.com/, 2025.

[7] K. Marky, K. Ragozin, G. Chernyshov, A. Matviienko, M. Schmitz, M. Mühlhäuser, C. Eghtebas, and K. Kunze, ""nah, it's just annoying!" a deep dive into user perceptions of two-factor authentication," *ACM transactions on computer-human interaction*, vol. 29, no. 5, pp. 1–32, 2022.

[8] J. Weidman and J. 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.

[9] L. Lassak, A. Hildebrandt, M. Golla, and B. Ur, ""it's stored, hopefully, on an encrypted server": Mitigating users' misconceptions about {FIDO2} biometric {WebAuthn}," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 91–108.

[10] T. Wei, D. Wang, Y. Li, and Y. Wang, ""who is trying to access my account?" exploring user perceptions and reactions to risk-based authentication notifications." in *32nd Annual Network and Distributed System Security Symposium, (NDSS'25)*, 2025.

[11] L. Lassak, E. Pan, B. Ur, and M. Golla, "Why aren't we using passkeys? obstacles companies face deploying FIDO2 passwordless authentication," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 7231–7248.

[12] J. H. Klemmer, M. Gutfleisch, C. Stransky, Y. Acar, M. A. Sasse, and S. Fahl, ""make them change it every week!": A qualitative exploration of online developer advice on usable and secure authentication," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023, p. 2740–2754.

[13] S. Ghorbani Lyastani, M. Backes, and S. Bugiel, "A systematic study of the consistency of two-factor authentication user journeys on top-ranked websites," in *30th Annual Network & Distributed System Security Symposium (NDSS'23)*, 2023.

[14] S. Sahin, B. Sahin, and F. Li, "Was this you? investigating the design considerations for suspicious login notifications," in *32nd Annual Network and Distributed System Security Symposium, (NDSS'25)*, 2025.

[15] J. Bonneau, C. Herley, P. C. Van Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," in *2012 IEEE symposium on security and privacy*. IEEE, 2012, pp. 553–567.

[16] X. Zhang, X. Zhang, B. Zhao, Y. Nan, Z. Liu, J. Chen, H. Zhou, and M. Yang, "Demystifying the (in) security of qr code-based login in real-world deployments," in *34th USENIX Security Symposium (USENIX Security 25)*, 2025.

[17] S. Ciolino, S. Parkin, and P. 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.

[18] S. G. Lyastani, M. Schilling, M. Neumayr, M. Backes, and S. Bugiel, "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.

[19] T. K. Yadav and K. E. Seamons, "A security and usability analysis of local attacks against FIDO2," in *31st Annual Network and Distributed System Security Symposium, (NDSS'24)*, 2024.

[20] A. T. Mahdad, M. Jubur, and N. Saxena, "Breaking mobile notification-based authentication with concurrent attacks outside of mobile devices," in *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, 2023, pp. 1–15.

[21] M. Jubur, P. Shrestha, N. Saxena, and J. Prakash, "Bypassing push-based second factor and passwordless authentication with human-indistinguishable notifications," in *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, 2021, pp. 447–461.

[22] Y. Yuan, N. H. Riche, N. Marquardt, M. J. P. Nicholas, T. Seyed, H. Romat, B. Lee, M. Pahud, J. Goldstein, R. S. Vishkaie, C. Holz, and K. Hinckley, "Understanding multi-device usage patterns: Physical device configurations and fragmented workflows," *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 2022.

[23] S. Zimmeck, J. S. Li, H. Kim, S. M. Bellovin, and T. Jebara, “A privacy analysis of cross-device tracking,” in *26th USENIX Security Symposium (USENIX Security 17)*, 2017, pp. 1391–1408.

[24] K. Solomos, P. Ilia, S. Ioannidis, and N. Kourtellis, “{TALON}: An automated framework for {Cross-Device} tracking detection,” in *22nd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2019)*, 2019, pp. 227–241.

[25] P. Hamilton and D. J. Wigdor, “Conductor: enabling and understanding cross-device interaction,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2014, pp. 2773–2782.

[26] W. Denniss, J. Bradley, M. B. Jones, and H. Tschofenig, “OAuth 2.0 Device Authorization Grant,” RFC 8628, Aug. 2019. [Online]. Available: https://www.rfc-editor.org/info/rfc8628

[27] Yubico, “Security key by yubico,” 2021.

[28] P. Kasselman, D. Fett, and F. Skokan, “Cross-Device Flows: Security Best Current Practice,” Internet Engineering Task Force, Internet-Draft draft-ietf-oauth-cross-device-security-12, Sep. 2025, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-oauth-cross-device-security/12/

[29] H. Nissenbaum, “Privacy as contextual integrity,” *Wash. L. Rev.*, vol. 79, p. 119, 2004.

[30] “European parliament and the council of the european union. general data protection regulation,” 2016.

[31] “California state legislature. california consumer privacy act,” 2018.

[32] P. Grassi, E. Newton, R. Perlner, A. Regenscheid, W. Burr, J. Richer, N. Lefkovitz, J. Danker, Y.-Y. Choong, K. Greene, and M. Theofanos, “Digital identity guidelines: Authentication and lifecycle management,” 2017-06-22 2017.

[33] I. P. T. ENTERPRISE, “Nist privacy framework: A tool for improving privacy through enterprise risk management, version 1.0,” 2020.

[34] D. K. Citron and D. J. Solove, “Privacy harms,” *BUL Rev.*, vol. 102, p. 793, 2022.

[35] “Symantec sitereview,” https://sitereview.bluecoat.com/#/, 2024.

[36] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczyński, and W. Joosen, “Tranco: A research-oriented top sites ranking hardened against manipulation,” in *Proceedings of the 26th Annual Network and Distributed System Security Symposium, (NDSS’19)*, Feb. 2019.

[37] B. Glaser and A. Strauss, *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine, 1967.

[38] N. McDonald, S. Schoenebeck, and A. Forte, “Reliability and inter-rater reliability in qualitative research: Norms and guidelines for cscw and hci practice,” *Proceedings of the ACM on human-computer interaction*, vol. 3, no. CSCW, pp. 1–23, 2019.

[39] P. Markert, L. Lassak, M. Golla, and M. Dürmuth, “Understanding users’ interaction with login notifications,” in *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–17.

[40] “Prolific - how to use custom screening to recruit specific participants.” https://researcher-help.prolific.com/en/article/6bad1f, 2025.

[41] S. Johnny, *The Coding Manual for Qualitative Researchers*. SAGE, 2016.

Appendix A
Questionnaire for Our User Study

Dear Participant:

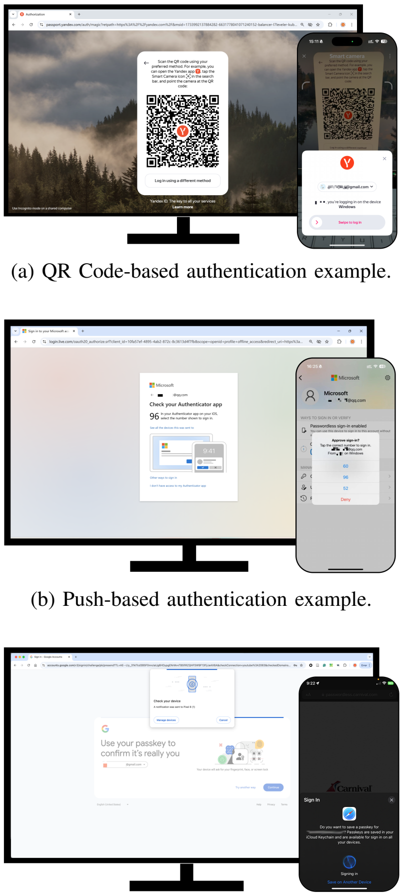
Thank you for participating in our study. This research focuses on cross-device authentication. Your participation is invaluable to us. In this study, you will be engaged in the following activities:

1) Viewing Examples of Cross-Device Authentication Methods: You will be presented with example images and videos of cross-device authentication methods to gain a more intuitive understanding.
2) Completing a Questionnaire: You will be invited to complete a questionnaire to share your opinions and experiences regarding cross-device authentication.

We assure you that your personal information will be strictly confidential. All collected data will be anonymized and used solely for the purposes of this study. Your participation is entirely voluntary, and you have the right to withdraw from the study at any stage without any penalty or adverse consequences.

The following three images are examples of cross-device authentication methods, which are QR code-based authentication, push-based authentication, and WebAuthn (Figure 5). Please review the images and answer the following questions.

- Fig 5a: An example of QR Code-based authentication.
- Fig 5b: An example of Push-based authentication.
- Fig 5c: An example of WebAuthn authentication.



(a) QR Code-based authentication example.

(b) Push-based authentication example.

(c) WebAuthn authentication example.

Fig. 5: Three Cross-device authentication examples.

Part 1

Q1. Which of the following login methods have you used? [Multiple choice]

☐ QR Code-based authentication (Figure 5a)
☐ Push-based authentication (Figure 5b)
☐ Webauthn authentication (Figure 5c)

Q2. The above login methods all use a logged-in mobile phone to authorize login on websites from another device, known as cross-device authentication. How often do you use cross-device authentication?

○ Almost daily ○ Several times a week
○ Several times a month ○ Other lower frequency

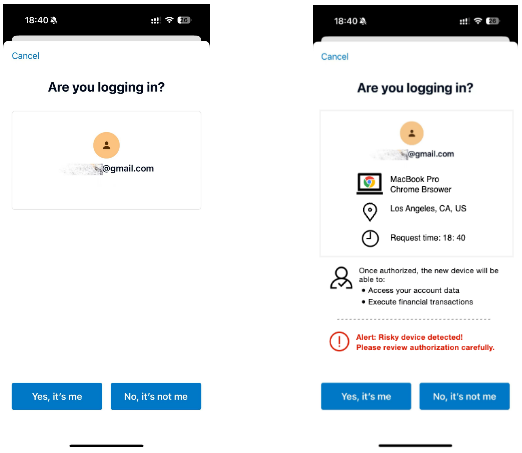
Part 2

Please watch the demo video[3] to understand how cross-device authentication works and answer the following questions about it.

Part 2-1. Pre-authentication

Q3. Below are two designs of pre-authentication workflow (Figure 6). Which one do you expect (prefer) to use? (The design diagrams are provided below.)

[3] The demo video is available at an anonymous link: https://imgur.com/a/OKIsEh4.



18:40

Cancel

Are you logging in?

[redacted]@gmail.com

Yes, it's me

No, it's not me

(a) Design A.

18:40

Cancel

Are you logging in?

[redacted]@gmail.com

MacBook Pro
Chrome Brsower

Los Angeles, CA, US

Request time: 18: 40

Once authorized, the new device will be able to:

- Access your account data
- Execute financial transactions

Alert: Risky device detected!
Please review authorization carefully.

Yes, it's me

No, it's not me

(b) Design B.

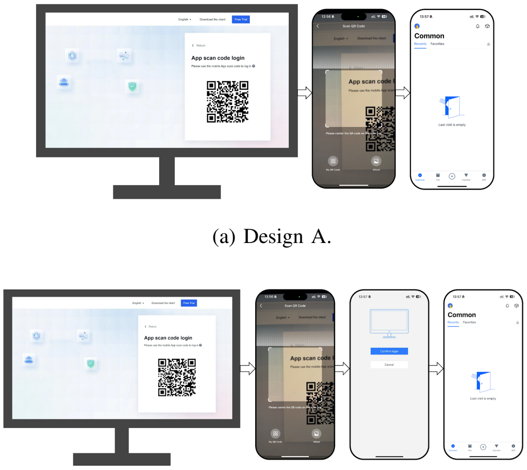
Fig. 6: Two designs in the pre-authentication stage.

○ Fig 6a. Doesn't provide any device and activity information before authorization.

○ Fig 6b. Provides information about the device to be authorized and activity information before authorization.

Q4. Why did you choose the selected method?________________

Part 2-2. During-authentication



(a) Design A.

(b) Design B.

Fig. 7: Two designs in the during-authentication stage.

Q5. Below are two designs of during-authentication workflow (Figure 7). Which one do you expect (prefer) to use? (The design diagrams are provided below.)

○ Fig 7a. Completes the login immediately after scanning the QR code without requiring user consent.

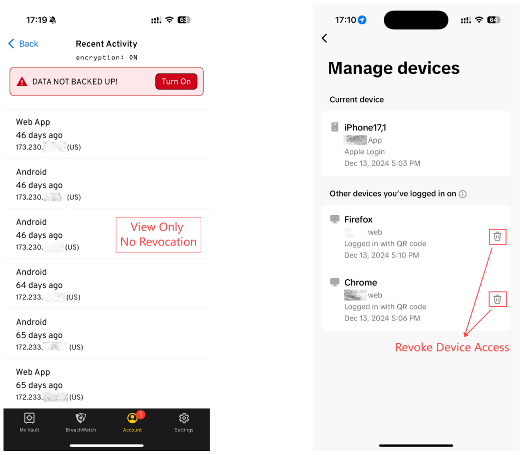
○ Fig 7b. Requires users to grant or deny authorization before logging in on another device.

Q6. Why did you choose the selected design?________________

Part 2-3. Post-authentication

Q7. Below are two approaches for managing account sessions (Figure 8). Which one do you expect (prefer) to use? (The design diagrams are provided below.)

○ Fig 8a. Allows users to only view the currently logged-in sessions or devices.



17:19

Back

Recent Activity

encryption: ON

DATA NOT BACKED UP! Turn On

Web App
46 days ago
173.230.[redacted] (US)

Android
46 days ago
173.230.[redacted] (US)

Android
46 days ago
173.230.[redacted] (US)

View Only
No Revocation

Android
64 days ago
172.233.[redacted] (US)

Android
65 days ago
172.233.[redacted] (US)

Web App
65 days ago
172.233.[redacted] (US)

(a) Design A.

17:10

Manage devices

Current device

iPhone17,1
[redacted] App
Apple Login
Dec 13, 2024 5:03 PM

Other devices you've logged in on

Firefox
[redacted] web
Logged in with QR code
Dec 13, 2024 5:10 PM

Chrome
[redacted] web
Logged in with QR code
Dec 13, 2024 5:06 PM

Revoke Device Access

(b) Design B.

Fig. 8: Two designs in the post-authentication stage.

○ Fig 8b. Allows users to view and log out any active sessions or devices in addition to viewing them.

Q8. Why did you choose the selected design?________________

Q9. Do you think these new features (the latter options in the three questions above, including Figure 6b, Figure 7b and Figure 8b) make cross-device authentication more secure?

○ Yes, the new features improve security.

○ Yes, the new features slightly improve security.

○ No, the new features do not affect the security.

○ No, the new features make it less secure.

Q10. Do you think these new features (the latter options in the three questions above, including Figure 6b, Figure 7b and Figure 8b) make cross-device authentication less usable (e.g., harder, slower, or more confusing), discouraging you from using it?

○ Yes, the new features harm usability, and I'd avoid it.

○ Yes, the new features somewhat harm usability, but I'd still want to use it.

○ No, the new features do not affect usability.

○ No, the new features increase usability.

Part 3

To support our analysis, please provide the following background information. The information we collect in this survey is for research purposes only. Your responses will be kept confidential and not disclosed to any third parties.

Q11. What is your gender?

○ Male ○ Female ○ Decline to say

Q12. What is your age range?

○ Below 18 ○ 18-30 ○ 31-45 ○ 46-60 ○ 61 above

Q13. What is the highest level of education you have completed?

○ No high school/Some high school/High school graduate

○ Some college – No degree

○ Associates (2-year degree) /Bachelor (4-year degree)

○ Graduate degree – Master, PhD, professional, medicine, etc

Q14. Which describes best regarding your professional background?

○ STEM (Science, Technology, Engineering, Mathematics)

○ Liberal arts

○ Other

Thank you! This ends our survey.

Appendix B
Results of Our User Study

A. Detailed Demographics

We specifically present the demographics data in our main user study and the validation study in Table V.

TABLE V: Participant demographics.

Demographic	**Subcategory**	**User Study (N = 100)**	**Validation Study (N = 10)**
Gender	Male	42	6
	Female	57	4
	Decline to say	1	0
Age	18-30	36	0
	31-45	37	7
	46-60	20	2
	61 above	7	1
Education Level	No/Some/Graduate high school	7	2
	Some college – No degree	11	2
	Associates / Bachelor	34	4
	Graduate degree	48	2
Professional Background	STEM Fields	55	4
	Liberal Arts	16	1
	Other	29	5

B. Codebook for Open-ended Responses

We present the codebooks for responses to open-ended questions in our user study and the validation study in Table VII and Table VI, respectively.

TABLE VI: Codebook for the Responses to Open-ended Questions in the validation study.

Right to Know		**Right to Consent**		**Right to Control**	
Code	Freq	Code	Freq	Code	Freq
Confirmation	8	Security	7	Revoke	6
Information	7	Control	5	Control	5
Security	5	User-friendly	2	Security	3
Transparency	4			User-friendly	2
Privacy	1			Visibility	1

C. User Rights Independence Tests

As shown in Table VIII, we demonstrate the independence tests analyzing the association between the user characteristics and their preferences for user rights. Yates' continuity correction was applied to 2×2 tables where expected cell frequencies fell below five.

TABLE VII: Codebook for the Responses to Open-ended Questions.

Code	Freq	Description	Example
Q4. Why did you choose the selected method? (right to know)			
Transparency	39%	The method increases transparency.	*“Providing information about the device and activity before authorization improves transparency and security.”*
Security	31%	The method improves security.	*“There’s more information to provide an extra layer of security.”*
Information	25%	Activity and device information.	*“Provides information about the device to be authorized is given.”*
Confirmation	21%	Confirm the device or identity.	*“It confirms more specifically that I’m the person logging in.”*
User-friendly	20%	The method is more user-friendly.	*“It is safer and more convenient.”*
Invalid	8%	Invalid answer or misunderstanding.	*[B] (only the option)*
Daily habit	5%	The daily habit of participants.	*“This is the method I have used or seen the most.”*
Privacy	4%	Worried about the privacy.	*“It can help identify who’s trying to access my privacy.”*
Not sure	2%	Not sure about the answer.	*“I’m not sure. It just seemed right.”*
Q6. Why did you choose the selected design? (right to consent)			
Security	55%	The design improves security.	*“Because it requires that the users approve or reject the authorization. It also fosters security.”*
Control	25%	Improves user access control.	*“It actually enhances user confidence and gives them more control over their security.”*
User-friendly	24%	The design is more user-friendly.	*“Easier and simple.”*
Invalid	5%	Invalid answer or misunderstanding.	*[B] (only the option)*
Daily habit	3%	The daily habit of participants.	*“I use it from time to time.”*
Cross-Device Authentication	3%	Log in on another device.	*“I expect to be asked whether or not I am granting use of the program on another device.”*
Transparency	2%	The design increases transparency.	*“Requiring users to grant or deny authorization before logging in ensures greater control, security, and transparency.”*
Q8. Why did you choose the selected design? (right to control)			
Revoke	30%	Revoke the existing sessions.	*“It allows you to log out any active users on your account that don’t belong there. It is more secure.”*
Control	26%	More control on sessions or devices.	*“I chose B because it provides greater control, enabling users to log out sessions or devices, improving security and privacy.”*
Security	25%	The design improves security.	*“In case a device is lost revoking access is way more secure.”*
User-friendly	20%	The design is more user-friendly.	*“It enhances security by providing a clear, user-friendly interface to monitor and manage devices connected to the account.”*
Visibility	7%	Visibility on sessions or devices.	*“Because it allows users to get the details of the devices where it’s logged on, and grant them the opportunity to either remove it or let it stay.”*
Protection	7%	Protection on privacy.	*“Because of data privacy and protection as well as easy to follow the arrest the hackers.”*
Invalid	5%	Invalid answer or misunderstanding.	*[B] (only the option)*
Verification	5%	Verification on sessions.	*“This is better, but there should be verification before critical actions can be taken.”*
Daily habit	3%	The daily habit of participants.	*“Seems more familiar to me.”*
More choices	3%	The design gives more choices.	*“I think this design is more flexible and gives more optionality.”*
Risk	2%	Security risks of accounts.	*“This feature is crucial for managing potential security risks, such as unauthorized access or forgotten logins on shared or public devices.”*

TABLE VIII: Chi-square Tests of Independence for User Characteristics and User Rights Preferences.

Variable	**User Rights**			**Overall Conclusion**
	Right to Know	**Right to Consent**	**Right to Control**	
Age	$\chi^2(1) = 0.54$, $p = .464$	$\chi^2(1) = 1.35$, $p = .245$	$\chi^2(1) = 0.08$, $p = .777$	No significant effect
Education	$\chi^2(1) < 0.01$, $p = 1.000$	$\chi^2(1) < 0.01$, $p = .986$	$\chi^2(1) = 2.46$, $p = .116$	No significant effect
Major	$\chi^2(1) = 0.10$, $p = .752$	$\chi^2(1) = 3.53$, $p = .060$	$\chi^2(1) = 0.18$, $p = .673$	No significant effect
Usage Frequency	$\chi^2(1) < 0.01$, $p = 1.000$	$\chi^2(1) = 0.36$, $p = .548$	$\chi^2(1) = 0.87$, $p = .351$	No significant effect