AD-DP: Device-Aware Anomaly Detection for Securing WebAuthn Passkey Authentication

Khin Win Myat Mon[1,a)] Shuji Yamaguchi[2,1,b)] Hidehito Gomi[2,c)]
Tetsutaro Uehara[3,d)]

Abstract: The evolution of FIDO2/WebAuthn from device-bound credentials to cloud-synced passkeys has opened new attack surfaces that traditional cryptographic checks cannot detect. In particular, attackers can exploit by using valid passkeys from unassociated or malicious devices and enabling **Man-in-the-Middle (MitM)** threats such as those shown in the CTAP Hijacking academic paper. This paper introduces **AD-DP (Anomaly Detection through Device Profiling)**, a server-side enhancement designed to close this security gap. AD-DP establishes a trusted relationship between users, credentials, and devices by profiling authentication behavior over time. It enables Relying Parties(RPs) to detect anomalous device usage before completing WebAuthn flows, providing an additional layer of real-time defense without modifying the standard protocol. We present the rationale for the design, the architectural components, and an evaluation plan. Our proposal addresses a critical gap in WebAuthn authentication by enabling servers to assess the legitimacy of the authenticating device, not just the credential, within existing authentication flows.

Keywords: WebAuthn, Passkeys, CTAP security, Hijacking, Device profiling, Anomaly detection, Cross-device authentication, Risk-based authentication, Verifiable credentials, FIDO2, Behavioral Biometric authentication, Synchronization threats, Man-in-the-Middle (MitM) attack

1. Introduction

Passwords remain the weakest link in digital security, implicated in most breaches through reuse, theft, and phishing attacks. To address these risks, the industry has moved towards passkeys, public-key-based authentication method built on the FIDO2/WebAuthn standard [1]. Passkeys are designed to eliminate shared secrets, simplify the login experience, and improve resistance to credential-based attacks.

Initially, passkeys were tied to a single physical device. However, to enhance usability across a user's device ecosystem, vendors have introduced synchronized passkeys—credentials that are securely synced through cloud-based management system like Apple's iCloud Keychain, Google's Password Manager, and Microsoft Password Manager [2]. These services allow users to log in seamlessly from multiple endpoints.

To support cross-device logins from different system such as sign-in on a laptop using a phone's passkey—modern systems employ secure pairing techniques using QR codes, Bluetooth Low Energy (BLE), or Near Field Communication (NFC)[3]. These transport mechanisms enable the primary device holding the private key to authorize the authentication request on another device, even when the devices are not directly synchronized. While this enhances accessibility, it also introduces new risks if any device in the trust chain is compromised[4].

While usability-driven, this evolution has expanded the attack surface. Even if the cryptographic signature in a WebAuthn response is valid, there is currently no way for the relying party (RP) to verify whether the authentication originated from a legitimate, user-associated device. This opens the door to identify cases where synced passkeys are used from compromised, cloned, or unrecognized devices. Recent academic work, the CTAP Hijacking attack [5] demonstrates how malware can hijack the Client-to-Authenticator Protocol (CTAP) to initiate passkey authentication flows without the user's consent [6]. Additionally, guidance from NIST highlights the danger of "unauthorized key use" in synchronized authenticators—where any device in the sync group may be exploited as an attack vector [7].

These threats reveal a critical limitation in current WebAuthn deployments: cryptographic validity does not imply device trustworthiness.

[1] Graduate School of Information Science and Engineering, Ritsumeikan University, Osaka, Japan
[2] LY Corporation, Tokyo, Japan
[3] College of Information Science and Engineering, Ritsumeikan University, Osaka, Japan
[a)] khin@cysec.cs.ritsumei.ac.jp
[b)] shyamagu@lycorp.co.jp
[c)] hgomi@lycorp.co.jp
[d)] t-uehara@fc.ritsumei.ac.jp

To address this gap, we propose **AD-DP (Anomalous Detection – Device Profiling)**, a server-side security enhancement that adds device context and behavioral analysis to Relying Party(RPs) for security enhancement in passkey based system. AD-DP introduces a dynamic trust model where relying parties can evaluate whether a login attempt originates from a previously trusted device or one that behaves anomalously—before completing the WebAuthn flow. This paper outlines the architectural design, profiling methods, and risk scoring framework behind AD-DP, and presents a plan for evaluating its effectiveness through simulated attack scenarios.

2. Background

2.1 Passkey Authentication Evolution

Passkeys are the latest evolution in the FIDO2 family of standards, emerging from the progression of the Universal Authentication Framework (UAF) to the FIDO2 specification. FIDO2 combines the Client-to-Authenticator Protocol (CTAP) for device communication and the Web Authentication API (WebAuthn) for browser-server interactions, enabling passwordless authentication across modern platforms [8].

In the original model, FIDO2 credentials were device-bound—anchored to a hardware authenticator like a security key or platform module. Each passkey was unique to the device, stored securely, and non-exportable. The underlying assumption was that device possession implied user legitimacy.[9].

However, as vendors like Apple, Google, and Microsoft pursued usability improvements, passkeys evolved into cloud-synced credentials. Through Cloud Management Services (PMS), credentials are now synchronized across a user's trusted device ecosystem [3]. This shift supports seamless cross-device login but undermines the original device-bound trust model, introducing new vectors for credential misuse.

2.2 Device Identity and Trust Gaps

The move toward synchronized credentials created a new problem: Relying parties (RPs) have no native way to verify whether a valid credential is being used on a known or trusted device. Although proposals like the devicePubKey extension (recently renamed to supplementalPubKey) [10]aim to enable per-device cryptographic identifiers, implementation is still early, and support across platforms remains inconsistent.

As a result, many WebAuthn-based systems cannot differentiate between a legitimate user's laptop and a malicious emulator using the same synced passkey. This leaves device legitimacy, user intent, and session context unverified—an exploitable blind spot.

2.2.1 Real-World Vulnerabilities in Cross-Device Passkey Use

Multiple classes of vulnerabilities have been demonstrated in the context of cross-device passkey usage[11]:

- **CTAP Transport Attacks**: The CTAP Hijacking attack shows how malware can hijack the CTAP channel[5], allowing attackers to trigger authentication without user intent.
- **QR/Proximity Relay Attacks**: Threat actors can intercept or spoof QR-code-based logins or nearby BLE/NFC communications—relaying authentication attempts from an attacker-controlled device[4].
- **Session Confusion**: Since credentials are valid across devices, a relying party may accept a legitimate passkey without knowing if the session or platform has changed[12].
- **User Intent Ambiguity** CTAP has no built-in mechanism to determine whether the user explicitly initiated the login—malicious software can impersonate user actions.

These issues demonstrate that valid cryptographic signatures alone are insufficient for ensuring trust in a synchronized credential world. As highlighted by NIST [7], any device within a passkey sync group becomes a potential attack vector if compromised.

2.3 Problem Statement

Attack Model - QR code CTAP Hijacking attack[6] Recent academic study, specifically targets QR code-based passkey authentication in cross-ecosystem scenarios, exploiting CTAP (Client-to-Authenticator Protocol) communication through Man-in-the-Middle (MitM) techniques. This attack vector is particularly dangerous in hybrid authentication workflows where users authenticate on one device (mobile) to access services on another device (desktop/laptop) via QR code scanning.

The attack sequence unfolds as follows:

(1) An attacker deploys a phishing website that mimics a legitimate service.
(2) In parallel, the attacker initiates a real authentication session on the target website using their own device.
(3) The legitimate site's QR code is intercepted and proxies through the phishing site.
(4) The victim scans the malicious QR code, believing they are authenticating to the phishing service.

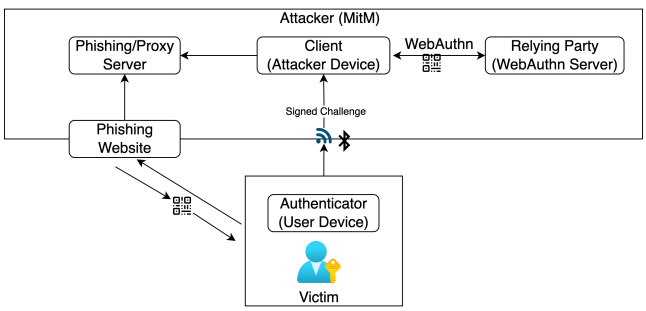

Fig. 1: Attack Model

(5) The victim's authenticator completes the CTAP authentication ceremony, but the active session belongs to the attacker's device on the real site.

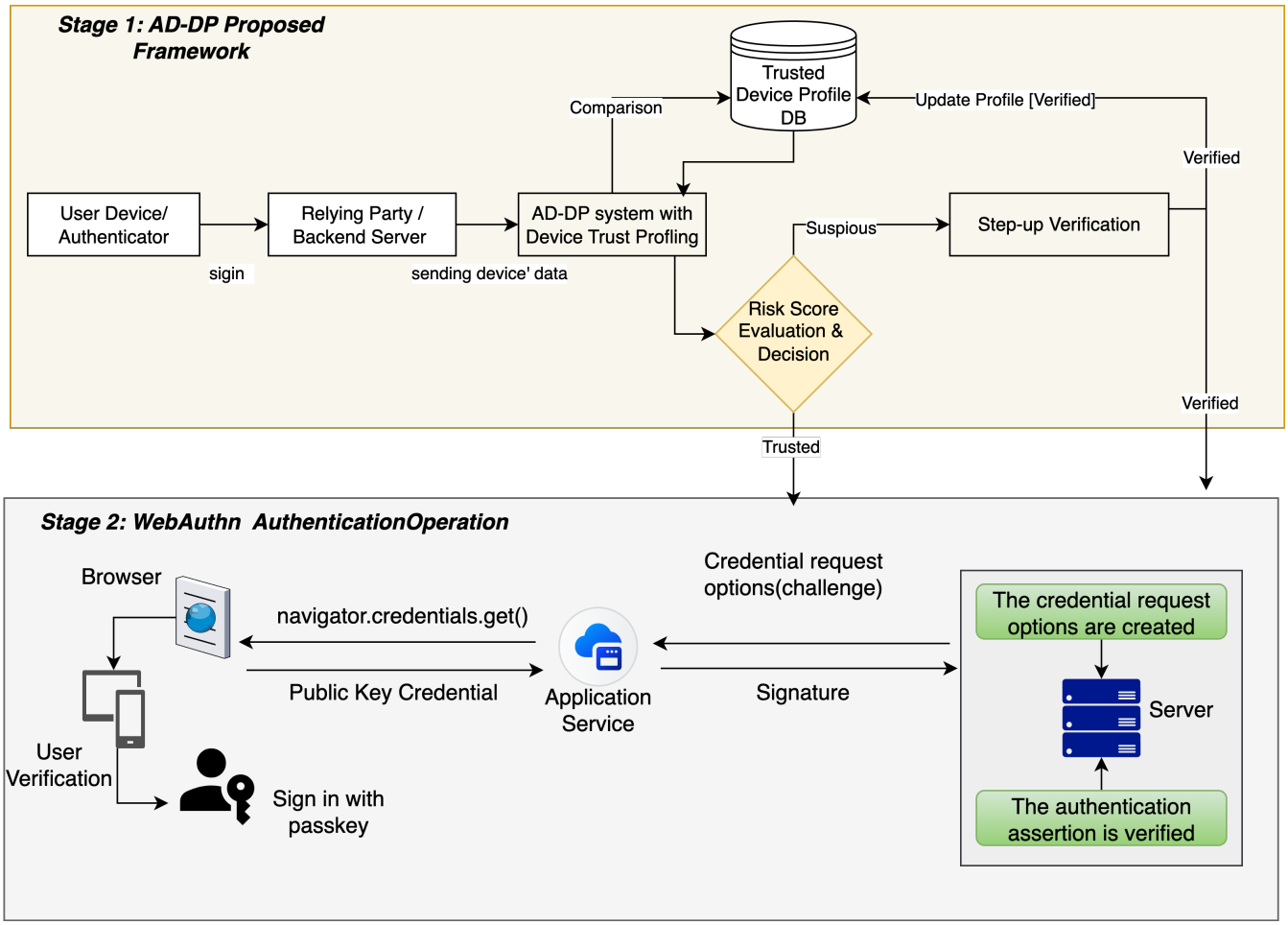

Fig. 2: AD-DP system flow chart

While CTAP correctly validates cryptographic signatures, it cannot verify whether the device requesting authentication should legitimately possess the passkey. In synchronized passkey environments, credentials can be accessed across multiple devices, leaving protocols blind to whether the authentication request originates from an authorized or anomalous device.

This attack scenario demonstrates that device substitution attacks can succeed without breaking cryptography, exposing a serious weakness in relying solely on cryptographic validation. Current security measures focus primarily on user awareness and phishing resistance, but lack technical mechanisms to detect protocol-level manipulation during live authentication ceremonies.

3. Contribution & Scope

This paper proposes **AD-DP (Anomaly Detection through Device Profiling)**, a server-side security architecture designed to detect untrusted or malicious device usage during WebAuthn-based passkey authentication, specifically addressing the blind spot in cross-device sign-ins as shown in Figure 2.

Our key contributions are:

- **Identification of a critical security gap** in WebAuthn and passkey flows, where relying parties (RPs) cannot distinguish between previously trusted devices and new/unassociated or potentially malicious devices, especially under CTAP hijacking [6].
- **A User–Credential–Device trust model** maintained entirely server-side, which links device identities and behaviors to WebAuthn credentials without modifying client-side authenticators or browser standards.
- **A layered device profiling system** combining as shown in Table 1:
  (1) **Authenticator Metadata(from WebAuthn Ceremony)**AAGUID, Transport type, Credential type & attestation
  (2) **Device Fingerprint(passively collected via browser**CPU,GPU renderer string, OS, Browser, Screen Resolution
  (3) **Behavioral & Timing Signals(during authentication)** CTAP response timing,flow sequences resulting in a per-device behavioral baseline that evolves across multiple sessions.
- **An anomaly detection pipeline** that scores authentication attempts based on deviation from learned device profile baselines, enabling dynamic classification of sign-in attempts as trusted or anomalous.
- **Support for verifiable credentials (VCs)**as an

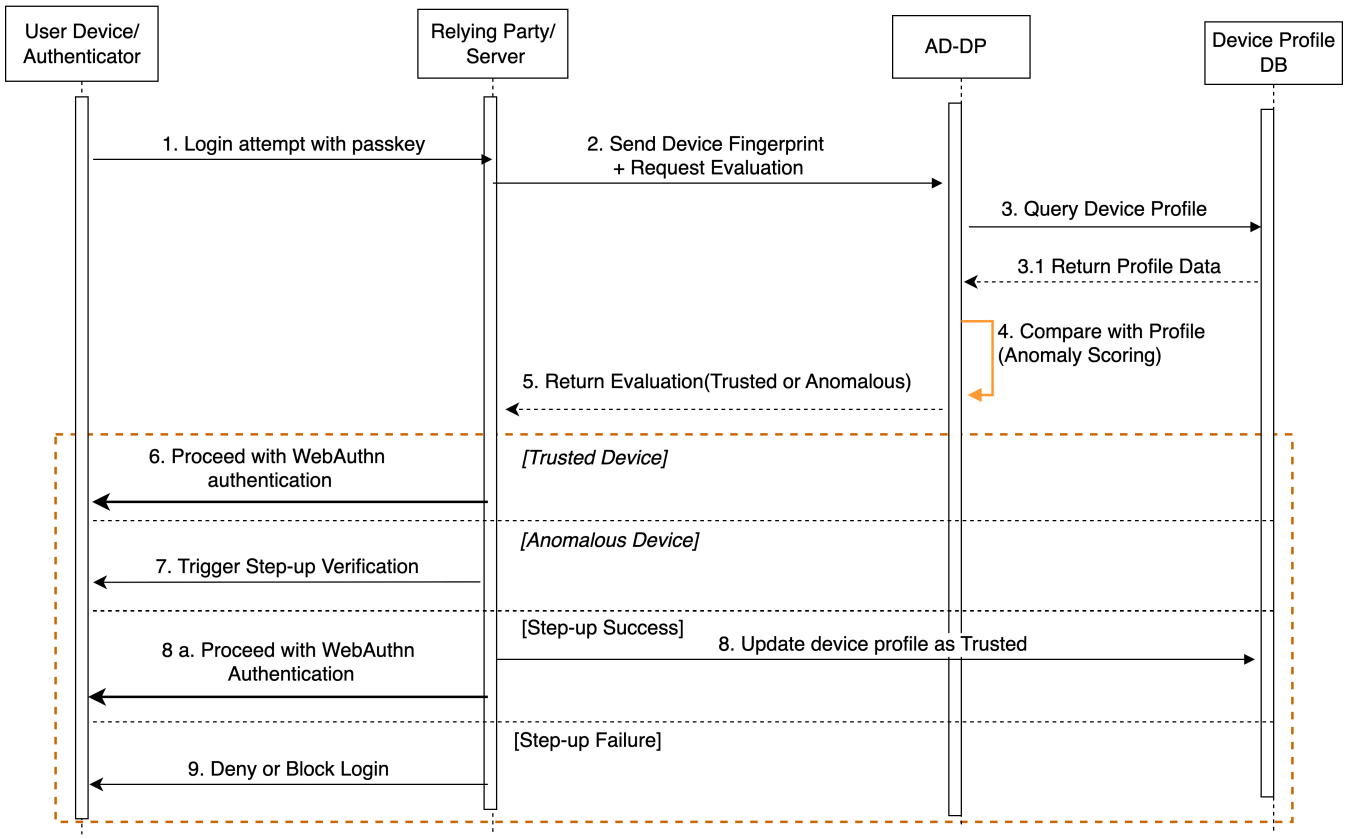

Fig. 3: AD-DP Sequence Diagram

optional fallback identity assertion mechanism during high-risk or uncertain device login attempts, offering privacy-preserving user attestation when behavioral patterns deviate

This paper focuses on the design, architecture, and technical feasibility of AD-DP framework. Full implementation, parameter tuning, and empirical evaluations under attack simulations are left for future work.

4. Proposed Method

4.1 Architecture Overview

AD-DP (Anomaly Detection through Device Profiling) is a server-side security enhancement designed to close the device trust gap in WebAuthn-based passkey authentication (shown in Figure 2). Unlike traditional implementations that treat all valid WebAuthn assertions equally, AD-DP builds **User–Credential–Device trust relationships** by associating each successful authentication with a **device profile** stored in a server-managed database as shown in Figure 4.

Each WebAuthn interaction is logged and used to construct or evaluate a device profile tied to the below identifier:

- **User ID** - A server-assigned unique identifier representing the user account during WebAuthn credential registration.
- **Credential ID** - A random byte string identifies the user's specific passkey, generated during WebAuthn credential registration.
- **Device profile** - A server maintained structure that represents devices' behavior data for authentication.

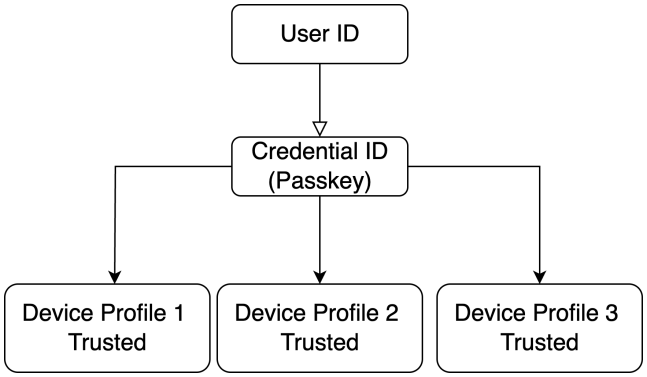

Fig. 4: Device Profile

This allows AD-DP to continuously refine trust relationships across devices and detect unauthorized or previously unseen devices attempting to use legitimate credentials.

4.2 Device Profiling Methodology

AD-DP uses a hybrid profiling approach combining WebAuthn Metadata, passive fingerprinting, and timing-based behavior.

The table as shown in Table 1 summarizes the minimal, stable, and privacy-conscious signals used to construct device profiles:

Table 1: Device Profile: Key Features

Category	Key Features	Purpose
Authenticator Metadata	AAGUID, transport type (USB, NFC, BLE), attachment mode (platform/roaming)	Identifies authenticator class & interface
Device Fingerprint	CPU architecture, GPU renderer, OS version, screen resolution, browser engine	Provides stable hardware/software identity
Behavioral Timing	CTAP response latency, sequence timing, UI interaction delays	Adds resilience against spoofing/fabrication

Table 2: Signal Weight Configuration

Signal Group	Feature Example	Weight w_i	Deviation Scoring Example
Authenticator Metadata	AAGUID mismatch	25	Full mismatch = 1.0
Transport Layer Info	Change in USB/NFC/BLE	15	Unseen transport = 1.0
Device Fingerprint	GPU or OS version change	15	Partial match = 0.5
CTAP Timing Behavior	Latency deviation	20	2x baseline timing = 0.7
Authentication Sequence	Auth flow (QR → BLE) change	25	Unfamiliar sequence = 1.0
	→ Max Score	**100**	

All data is collected passively, anonymized where appropriate, and stored under hashed user-credential-device mappings.

4.3 Risk Scoring

To assess the trustworthiness of a device during authentication, AD-DP computes a composite risk score based on deviations from the previously established device profile baseline.

Each signal group is assigned a **weight** and scored for deviation severity.
The final risk score R is calculated using the following formula:

$$R = \sum_{i=1}^{n} w_i \cdot d_i$$

Where:

- w_i = weight of the i^{th} signal group
- d_i = normalized deviation score (0 to 1) of that group
- $R \in [0, 100]$

4.3.1 Signal Weight Configuration

The table as shown in Table 2 summarizes as sample:

These weights can be adjusted by the relying party (RP) depending on their risk posture and observed false positive tolerance.

4.3.2 Decision Logic

The risk scoring mechanisms in AD-DP is designed to be configurable based on operational context and risk appetite.

For the purposes of this proposal, we define three threshold ranges to illustrate decision-making logic. These values are not fixed and would be empirically tuned during system evaluation and deployment.

- 0–40 = trusted
- 41–70 = uncertain
- 71–100 = high risk

The modular scoring framework allows relying parties to adjust both signal weights and risk thresholds depending on policy, user behavior patterns, and acceptable false positive rates.

4.4 AD-DP Framework Sequence Flow

AD-DP integrates with existing WebAuthn without protocol modifications showing in Fig. 3:

(1) User initiates passkey authentication - Collect metadata, fingerprint and behavioral timing (Learning Phase)
(2) Device Profile Lookup - Match User ID + Credential ID + Fingerprint
(3) Baseline comparison (Detection Phase) - Evaluate signal deviation and score
(4) Decision logic (Decision phase)
- If trusted : Proceed WebAuthn Authentication ceremony
- If anomalous: Request Step-up authentication via Verifiable credentials(VCs).
- If high risk : Deny or alert

4.5 Optional Step-up Authentication

When a session falls into the intermediate risk range (41–70), **AD-DP** provides the option to trigger a step-up authentication mechanism. The specific method is left to the Relying Party (RP) to define based on their security posture and ecosystem compatibility.

One recommended approach is the use of Verifiable Credentials (VCs) W3C compliant cryptographic identity attestations that can be presented from digital wallets[13]. Other alternatives (e.g., OTP, email verification, or biometric reconfirmation) may also be supported by the RPs.

- If Step-up Succeeds , the device is added/updated to the user's trusted baseline and proceed WebAuthn authentication ceremony
- If Step-up Fails, relying party blocks the login request. In this paper, VCs are used as the illustrative fallback mechanism, but are not required for our implementation or evaluation.

5. Discussion

5.1 Threat Model & Simulated Attack Scenario

Our method targets to detect simulated attacks where a valid passkey is used from an unassociated or spoofed device, such as in CTAP hijacking or credential syncing abuse.

The system assumes that while attackers may possess the credential (e.g., via sync or phishing), they do not have access to previously trusted device behavior. We simulate such threats by replaying valid credential responses from altered device environments and measuring the system's detection rate.

5.2 Evaluation Plan

To evaluate AD-DP's effectiveness, we will:

- Simulate authentication flows from both trusted and tampered devices
- Measure detection rates across different signal deviations and risk scores
- Assess false positives by replaying authentication from slightly altered but legitimate devices
- Validate that trusted profiles evolve without compromising detection accuracy

  The focus will be on precision, recall, and profile robustness—not on fallback performance or user friction at this stage.

5.3 Signal Drift & Legitimate Changes

Natural device changes (e.g., OS updates, browser version shifts) may cause small deviations. AD-DP is designed to tolerate benign drift through flexible scoring and optional step-up verification. If fallback succeeds, the system updates the profile to reflect the new configuration—supporting legitimate evolution over time.

6. Future Work

While we outlines the system design, scoring and decision conceptual model, we see it as a foundation for a broader, smarter verification layer in authentication systems. Several enhancements are planned to take the system from concept to a practical implementation, evaluation and adaptable security tool.

6.1 Advanced Extensions

Anomalous Access Detection with User Behavior Patterns

As part of our project, the Risk-based Multimodal FIDO Authenticator (RMFA) proposed in this research is unique in that it dynamically adjusts authentication levels according to risk and integrates keystroke authentication to optimize identity estimation and additional authentication during a session [14].

While the **AD-DP** Framework focuses on "device trustworthiness of synchronized passkeys," **RMFA** research focuses on "risk-based authentication using multi-factor signals," presenting a different approach to augmenting FIDO2.

Beyond baseline device profiling, future iterations of AD-DP will integrate the **RMFA**[14] framework to detect suspicious access patterns that may bypass or mimic legitimate credential use. These extensions aim to evolve AD-DP from a static device profiling model into a continuous, adaptive trust engine, enhancing the detection of sophisticated attacks without compromising usability.

7. Conclusion

In this paper, we introduced AD-DP, a server-side architecture that enhances passkey-based authentication by learning and evaluating device behavior over time. Instead of relying on static identifiers or generalized device fingerprints, AD-DP builds a trusted relationship between a user, their credential, and their device profile.

By combining device metadata, transport characteristics, and CTAP behavioral signals, the system detects when a valid passkey is used from an unrecognized or suspicious device—filling a critical gap in current WebAuthn deployments. Our risk scoring model classifies sessions in real time, with the flexibility to trigger fallback verification when needed.

While our current work focuses on design and feasibility, future phases will focus on full implementation, evaluation under simulated attack scenarios, and expansion into user behavior modeling and privacy-preserving learning methods.

AD-DP aims to provide relying parties with an adaptive, lightweight layer of trust—one that grows with the user, catches anomalous access, and preserves the simplicity of passkey authentication.

References

[1] FIDO Alliance: Multi-Device FIDO Credentials, *Whitepaper*, (online), available from ⟨https://fidoalliance.org/white-paper-multi-device-fido-credentials⟩ (2022).

[2] FIDO Alliance: Apple, Google and Microsoft Commit to Expanded Support for FIDO Standard to Accelerate Availability of Passwordless Sign-Ins, *News Center*, (online), available from ⟨https://fidoalliance.org/apple-google-and-microsoft-commit-to-expanded-support-for-fido-standard-to-accelerate-availability-of-passwordless-sign-ins/⟩ (2022).

[3] Andre Büttner, N. G.: Device-Bound vs. Synced Credentials: A Comparative Evaluation of Passkey Authentication, *Computer Sciences*, (online), DOI: 10.48550/arXiv.2501.07380 (2025).

[4] Casagrande, M.: Protocol-level Attacks and Defenses to Advance IoT Security, Theses, Sorbonne Université (2024).

[5] Marco Casagrande, D. A.: CTRAPS: CTAP Client Impersonation and API Confusion on FIDO2, *CTRAPS*, Marco Casagrande, (online), DOI: https://doi.org/10.48550/arXiv.2412.02349 (2024).

[6] D. Kim, J. Shin, G. R. and Choi, D.: HiPass: Hijacking CTAP in Passkey Authentication," in IEEE Access, vol. 13, pp. 92086-92101, 2025, *Applied Sciences*, Vol. 15, No. 8 (online), DOI: 10.1109/ACCESS.2025.3570377 (2025).

[7] David Temoshok, e. a.: Digital Identity Guidelines: Authentication and Authenticator Management, *NIST SP 800-63B-4*, (online), DOI: 10.6028/NIST.SP.800-63B-4 (2025).

[8] FIDO Alliance: User Authentication Specifications Overview, *blog*, (online), available from ⟨https://fidoalliance.org/specifications/⟩ (2022).

[9] Thurupati, S. C.: Passkeys and the Paradigm Shift in

Authentication: A Comprehensive Analysis of Phishing-Resistant IAM., *International Journal of Research in Computer Applications and Information Technology*, Vol. 7 (online), DOI: https://doi.org/10.5281/zenodo.14035089 (2024).

[10] Vincent: Device-Bound vs Synced Passkeys, *blog*, (online), available from ⟨https://www.corbado.com/blog/device-bound-synced-passkeys⟩ (2025).

[11] Barbosa, M., Boldyreva, A., Chen, S. and Warinschi, B.: Provable Security Analysis of FIDO2, Cryptology ePrint Archive, Paper 2020/756 (2020).

[12] Georgiadis, E.: FIDO2 Overview, Use Cases, and Security Considerations, PhD Thesis, Athens university (2023).

[13] Vincent: Digital Credentials Passkeys: How they align differ, *blog*, (online), available from ⟨https://www.corbado.com/blog/digital-credentials-passkeys⟩ (2025).

[14] Yamaguchi, S., Khin, W. M. M., Gomi, H., Kominami, K. and Uehara, T.: Toward a Risk-based Multimodal FIDO Authenticator: Proposal and Empirical Evaluation of Keystroke Dynamics-Based Authentication, *Proc. of Computer Security Symposium* (2025).