Towards User-Recognizable Device Identifiers

Jianyi Wang
University of Wisconsin–Madison
jwang2682@wisc.edu

Waleed Arshad
University of Wisconsin–Madison
warshad@wisc.edu

Alaa Daffalla
Cornell University
alaadaffalla@cs.cornell.edu

Cody Sondgeroth
University of Wisconsin–Madison
csondgeroth@wisc.edu

Rahul Chatterjee
University of Wisconsin–Madison
rahul.chatterjee@wisc.edu

Abstract

Account login history interfaces are designed by online services to help users detect unauthorized access to their accounts. However, current implementations provide unreliable and easily spoofed information such as device models and approximate cities or provinces derived from IP addresses. These cues are especially unreliable in *interpersonal attacks* (IA), where attackers personally know the victim (e.g., intimate partners, family members, or colleagues) and often share similar devices, networks, or locations. We introduce *User-Recognizable Device Identifiers (URIDs)* — unspoofable, unlinkable device identifiers designed to help users recognize logins from their own devices. URIDs are attested by device hardware to prevent spoofing and represented as AI-generated images that users can easily remember and recognize. Embedded into standard app or web interfaces, these images allow users to visually verify their logins in the account login history without revealing device identity across services. We implement URIDs on the Android mobile platform and evaluate their effectiveness in a three-week within-subjects study with 59 participants. In this study, URIDs were associated with a statistically significant increase in overall login-detection performance (F_1, Holm-adjusted $p = 0.048$), with the largest directional gains in the most difficult scenarios where attacker logins mimic legitimate sessions using identical user-agent strings and IP addresses — conditions common in interpersonal attack cases.

Keywords

device identifiers, account security, interpersonal attacks, trusted execution environment, visual recognition memory, user study

1 Introduction

Detecting unauthorized logins remains a persistent challenge, particularly in the context of *interpersonal attacks (IA)*, where the adversary is someone close to the victim (e.g., an intimate partner, family member, or colleague). Intimate partner violence (IPV) is widespread — global estimates indicate that more than one quarter of ever-partnered women have experienced physical or sexual violence by a partner [50, 55, 60]. Account intrusions in this setting are especially difficult to diagnose because an attacker may share the victim's physical environment, network, and device types.

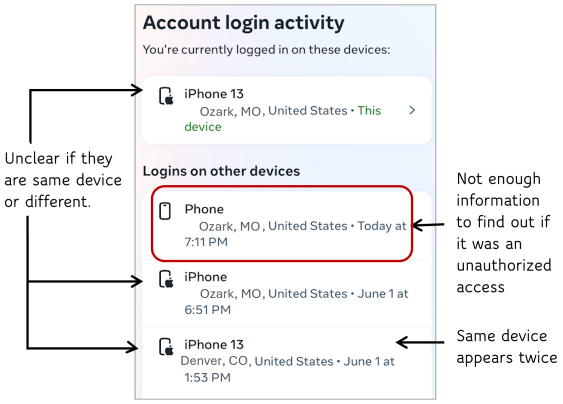


Figure 1: An illustrative example of the Facebook ALH interface showing incomplete and potentially misleading information, which makes it difficult for users to distinguish legitimate logins from suspicious ones.

In practice, users rely on the *account login history (ALH)* to identify suspicious activity. These interfaces present recent login events along with contextual metadata such as time, location, and device information (see Fig. 1). However, prior work shows that users struggle to interpret these signals effectively [20]. The core issue is that existing cues are either unreliable or easily manipulated. Users have limited ability to recall precise login times [19]; privacy relays can obscure network location [9]; and client-supplied device properties can be altered [7]. In IA scenarios, these signals degrade further: an attacker operating from the same network or using a similar device leaves little observable difference from legitimate activity.

Although modern devices provide strong hardware-backed identifiers (e.g., IMEIs, MAC addresses, or keys held in a Trusted Execution Environment (TEE) [57]), these are intentionally restricted by operating systems to prevent cross-application tracking [6, 10]. Moreover, even if accessible, such identifiers are not meaningful to users and therefore provide little value in user-facing interfaces like the ALH.

To address this gap, any practical solution must satisfy three requirements: it must be (a) *unspoofable* to resist adversaries, (b) *privacy-preserving* to prevent cross-service tracking, and (c) *user-recognizable* to support human decision-making. These requirements are inherently in tension. For example, globally unique identifiers risk linkability across services, while human-friendly representations are often easier to imitate.

Proceedings on Privacy Enhancing Technologies YYYY(X), 1–18

https://doi.org/XXXXXXX.XXXXXXX

We propose *User-Recognizable Device Identifiers (URIDs)* to satisfy all three requirements at once. URIDs are derived from hardware-backed keys within the TEE (Section 4), ensuring resistance to spoofing. To preserve privacy, identifiers are generated per *device–identity pair* (e.g., where the identity is an email address or a phone number), so that services cannot link a device across accounts unless they already share the same identity information.

To make identifiers usable, we convert them into AI-generated images (or other perceptual representations) that are periodically shown to users in familiar interface elements (e.g., lock screens or app splash pages). During login, the device shares its URID with the service, which records and displays it in the ALH. Users can then identify legitimate sessions by recognizing familiar URID images, while unfamiliar or missing identifiers can be flagged as suspicious. This approach leverages recognition memory, allowing users to judge a presented stimulus as familiar without having to reproduce an identifier from memory [46].

Prior work demonstrates a substantial capacity for long-term recognition of pictures and object details [15, 47]. This evidence motivates the use of visual identifiers, although recognizing repeatedly exposed URIDs in an account-security interface is a distinct task that we evaluate empirically.

We implement our URID scheme on Android devices using hardware-backed TEEs and generate images using the open-source SDXL-Turbo model. Image generation can be performed offline and cached, avoiding runtime overhead on user devices. To evaluate URIDs, we conducted a three-week within-subjects study with 59 participants. Participants interacted with ALH interfaces under conditions with and without visual identifiers. In this study, URIDs were associated with higher overall F_1 and the largest directional gains in the most challenging attack scenario, where conventional cues fail; participants also showed strong recognition of identifiers over time.

Contributions.

- We introduce URID, an unspoofable, unlinkable, and user-recognizable device identifier that enables users to distinguish legitimate logins from unauthorized ones in the ALH, particularly in IA scenarios.
- We design and implement a URID generation and verification system on Android using TEEs, and represent identifiers as AI-generated images to support intuitive recognition.
- We conduct a three-week user study with 59 participants demonstrating that URIDs improve login discrimination and are reliably recognized over time without explicit memorization.

2 Background and Related Work

Automated Compromise Detection. Service providers (SPs) detect compromised social-network accounts from changes in content, behavior, and social activity [25, 45, 56, 58, 63]. Thomas et al. [56], for example, analyzed 8.7 billion tweets and identified approximately 14 million affected accounts, many linked to phishing and malware spreading through the social graph. VanDam et al. [58] characterized self-reported Twitter compromises, while Ruan et al. [45] modeled Facebook users' clickstream behavior. Such methods help providers flag suspicious activity but do not give users trustworthy, intelligible evidence for attributing a login to a device.

User-Facing Compromise Diagnosis. Users often discover hijacking and account compromise through visible consequences, such as messages sent from their accounts [8, 54]. Login notifications can support earlier diagnosis, yet users remain uncertain about suspicious incidents and need more help responding to malicious sign-ins [37, 44].

In a user study simulating real-world account compromise via adversarial passkeys, Daffalla et al. [21] similarly find that users are often confused about login alerts and are hesitant to click on email links. They also find that service-provided remediation wizards are misleading and often fail to address the source of the compromise. This is consistent with prior work on online remediation guidance [36, 39] and users' responses to it [31, 49]. Research on security warnings more broadly finds that users act on them only when the warning explains the risk clearly and offers a concrete next step [1, 26] — requirements that login alerts and remediation flows largely fail to meet.

Account security interfaces (ASIs) [20], including device lists and session and activity logs, provide more context than individual notifications because they list all accesses to an account typically along with metadata such as the location, and date and time of access. Daffalla et al. [20] found that ASIs can aid compromise diagnosis but are difficult to interpret by users and may display attacker-manipulated metadata. Their work surfaces access hiding and spoofing attacks that undermine the integrity of ASIs.

In follow-up work, Bhattacharya et al. [14] conducted a survey of ASIs across 100+ services and found that 29 of 100 services provided no way to distinguish account accesses and that 67% of those conveying device or location information can be spoofed.

Thus, an ASI needs to carry information that is both manipulation-resistant and meaningful to its user. URIDs provide a recognizable representation derived from an attested device key, subject to our threat-model assumptions.

Accountable and private access logs. Many works in the literature have discussed the security and integrity of system and audit logs [13, 18, 30, 32, 34, 53]. Comparatively little, however, has explored access logging, and private access logging in particular.

The earliest attempt in this direction was account access graphs which model relationships among accounts, credentials, and access events [29], later extended to reason about attacker strategies [11]. Moving from modeling to mechanism, Larch ensures that authentication leaves evidence in a user-decryptable log without revealing the relying party to the log service [22]; it does not aim to make individual devices recognizable.

Client-side encrypted access logging (CSAL) adds private, OS-attested device attribution to account logs [40]. CSAL protects integrity, privacy, and unlinkability, whereas URIDs address how users recognize an attested identifier. A combined design could encode a URID in CSAL's encrypted attribution data, though evaluating that composition remains future work.

Recognition-Based Interface Design. Recognition presents a candidate stimulus, whereas recall requires retrieving information without that cue [28, 46]; recognition can draw on familiarity and

contextual recollection [62]. Interfaces therefore commonly favor recognizable choices [17], and visual-memory studies show substantial capacity for retaining pictures and object details [15, 35, 47]. These findings motivate our hypothesis that prior exposure helps users recognize device images in an ALH, which we test empirically.

Several security mechanisms similarly exploit images. CEAL maps public strings to human-distinguishable visual key fingerprints for side-by-side comparison [12]. Anti-phishing systems use personalized imagery to authenticate current browser or domain interactions [23, 61], and graphical passwords require users to select secret images [24, 59]. All these methods require users to make a security decision within their regular workflow (e.g., while logging in or reading email). URIDs instead support delayed attribution of recorded logins in the ALH. This ensures that users are not making security decisions in a rush. These images are not secret.

3 Design Requirements of URID

We design URIDs to help users recognize their own devices in the ALH without creating a new tracking mechanism. This section first defines the attacker and the trusted components, then states the security and usability requirements that guide our design.

3.1 Threat Model and Trust Assumptions

We consider an attacker who has obtained the victim's account credentials and can complete a login from an attacker-controlled device despite any risk-based authentication. The attacker may manipulate client-supplied metadata, use a VPN or the victim's network, and observe identifiers, certificates, and protocol responses. Their goal is to make the resulting ALH entry appear to originate from one of the victim's devices. Because URIDs are public identifiers rather than credentials, seeing a victim's URID is within the threat model; producing an accepted response for that identifier from a different device should remain infeasible. Attacks performed from the victim's unlocked device are out of scope because they legitimately use that device's key and URID.

We trust the hardware-backed Keystore or TEE to protect device keys and attestation, and the operating system (OS) to authorize access to those keys. We also trust the Privacy Certifying Authority (Privacy CA) to certify only valid attested keys, the SP to verify certificates, signatures, and freshness, and the ALH to retain verified login records. Compromise of any of these components, including suppression or deletion of ALH entries, is out of scope. Subject to these integrity assumptions, the Privacy CA and SP are honest-but-curious and may try to link devices or identities from the protocol messages they observe.

Security Requirements. The threat model yields the following security requirements for a URID system: (a) *Verifiable*: An SP can verify that an identifier response was produced by a certified device key for the requested account identity, without learning the device's persistent hardware identity. (b) *Unforgeable*: An attacker operating another device cannot produce a response that the SP accepts as originating from the victim's device, except by breaking the assumed TEE/Keystore or certificate trust model. (c) *Fresh*: A response is bound to the SP's challenge and a fresh timestamp, preventing replay of an earlier valid response. (d) *Per-identity unique*: For a fixed account identity, distinct devices produce distinct identifiers except with negligible probability. This requirement is scoped to the device–identity pair rather than a globally stable hardware identifier. (e) *Unlinkable across identities*: An SP should not be able to infer that identifiers generated for different account identities belong to the same device. The Privacy CA may validate device attestations, but it must not disclose the persistent hardware identity to SPs. (f) *Non-secret*: An identifier may appear in an ALH or notification and therefore must not encode sensitive information such as biometrics or account credentials.

Usability Requirements. For URIDs to be effective, they must also satisfy the following usability requirements: (a) *User recognizable*: Given an identifier in an ALH, a user can recognize whether it belongs to a device they used recently without physical access to that device. (b) *Low overhead*: Issuance and verification are computationally efficient, and recognizing an identifier adds little cognitive burden to routine account review. (c) *Accessible*: The identifier can be represented through multiple modalities, including images, text, audio, or symbols. We evaluate the image representation in this work; other modalities are left for future work.

3.2 Design Rationale

These requirements rule out several naive choices. IMEIs, permanent hardware identifiers, and biometrics may be unique or difficult to forge, but they are sensitive and linkable. User-selected device names are recognizable but forgeable and non-unique. A signature on a random value can provide cryptographic authenticity, but an opaque signature is not user recognizable and a persistent device key can enable cross-service tracking.

We therefore derive a machine-recognizable identifier (MRID) from a key protected by the device's TEE, and use a Privacy CA to replace device-identifying attestation metadata with a certificate suitable for SP verification. Each identifier is derived for a device–identity pair, where the identity is an account credential such as an email address or phone number. Different identities on the same device consequently yield unrelated identifiers, while a single identity can produce a consistent identifier for verification. The implementation and protocol are described in Section 4.

Finally, we render the MRID as a user-recognizable representation. We use AI-generated images in this work because they provide a large space of distinct, perceptually meaningful stimuli, but the security properties come from hardware-backed attestation rather than from image secrecy. This separation lets the study evaluate recognition while the threat model makes clear which claims depend on the trusted hardware and certificate authorities.

4 Instantiation of URID

URID generation has two components: (a) generating machine-recognizable identifiers (MRIDs) that meet the security requirements (Section 3), and (b) converting an MRID into images that serve as the user-recognizable identifiers (URIDs).

We build on Android's Hardware-Backed Keystore [2] and Key Attestation framework [4]. The Keystore can bind cryptographic keys to secure hardware — either a TEE (e.g., Trusty TEE [5]) or a dedicated security module exposed through StrongBox [3]. When

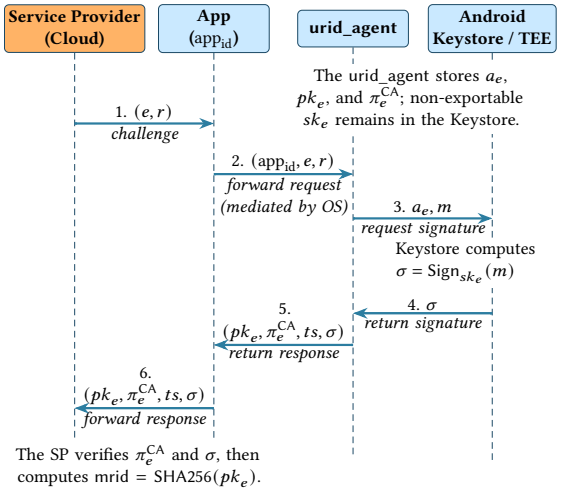


Figure 2: Online MRID protocol. The non-exportable key sk_e remains in the Android Keystore/TEE; the urid_agent invokes it through alias a_e. The initialization and certification steps precede this exchange and are not shown in this figure.

hardware-backed storage is available, key material can remain outside the Android OS; earlier Keystore designs without sufficient integrity protection have nevertheless been vulnerable to forgery [48]. Key Attestation provides signed evidence about where a key was generated and how it is protected, producing an attestation certificate chain that a verifier checks against a trusted root. SPs can verify this chain and use the attested public key to derive the cryptographic seed for the URID image. Fig. 8 provides screenshots of the user interface.

While this design ensures verifiability, it exposes the device's endorsement key (EK), a persistent identifier that can enable cross-service tracking. To preserve unlinkability, we introduce a *Privacy CA* that verifies hardware attestations and issues short-lived, unlinkable certificates that devices present to SPs, without revealing the EK. This approach maintains attestation integrity while ensuring that URIDs remain unspoofable, privacy-preserving, and unlinkable across services.

4.1 MRID Generation Process

We now specify how a device obtains and proves possession of an identity-specific key. Each device generates an independent key pk_e for each identity e of a user. We use a trusted Privacy CA [42] whose public key pk_{CA} is known to all SPs. There is an OS service called urid_agent (UA) that mediates between the SP and the device's Keystore. The signature and verification part of the URID protocol is shown in Fig. 2.

CA.Setup(1^λ). The Privacy CA generates its signing key pair (sk_{CA}, pk_{CA}). System parameters specify the signature scheme (e.g., ECDSA or EdDSA), a hashing scheme SHA256, a maximum timestamp age Δ, and an injective canonical encoding function encode for typed tuples. SPs trust pk_{CA} and maintain outstanding challenges.

UA.Init(e). With the user's approval, the urid_agent asks the hardware-backed Keystore to generate a key pair under a fresh alias a_e. The Keystore generates (sk_e, pk_e) and attestation evidence π_e^{Att}. It never exports sk_e; the urid_agent stores only the mapping $e \mapsto a_e$ and the public key pk_e. Each device runs Init independently for each identity.

CA.Issue(pk_e, π_e^{Att}). The urid_agent sends pk_e and the attestation evidence π_e^{Att} obtained from the Keystore to the Privacy CA to request a certificate. The Privacy CA validates the attestation chain π_e^{Att} and required security properties (e.g., that the key resides in a valid TEE and has not been revoked). If validation succeeds, it returns a (short-lived) certificate π_e^{CA} binding pk_e to those properties while omitting device-identifying attestation metadata. The urid_agent stores π_e^{CA} and repeats this request to renew it before expiration.

SP.GetSignature. The SP samples a random challenge r, records (app_{id}, e, r) as outstanding, and sends (e, r) to the app on the device. After explicit user approval, the app forwards (app_{id}, e, r) via the OS to the urid_agent. As we trust the OS, the urid_agent can verify that the message is from a valid app with app ID app_{id}. The urid_agent retrieves a_e corresponding to the identity e from its database; if no entry exists, it creates one by running UA.Init(e). It then reads the current timestamp ts, sets

$$m = \text{encode}(\texttt{URID-GetSignature-v1}, app_{id}, e, r, ts),$$

and asks the Keystore to sign m using the key with alias a_e. The Keystore computes $\sigma \leftarrow Sign_{sk_e}(m)$ internally and returns σ. The urid_agent forwards this signature along with pk_e, π_e^{CA}, and ts as $\rho = (pk_e, \pi_e^{CA}, ts, \sigma)$ to the app, which forwards it to the SP, as shown in Fig. 2.

SP.Verify(e, app_{id}, r, ρ). The SP parses $\rho = (pk_e, \pi_e^{CA}, ts, \sigma)$ and accepts only if: (1) π_e^{CA} is valid, unexpired, and binds pk_e; (2) (app_{id}, e, r) is outstanding; (3) ts is within Δ of the SP's clock; and (4) σ verifies under pk_e on encode(URID-GetSignature-v1, app_{id}, e, r, ts). If all checks pass, it atomically removes (app_{id}, e, r) from the outstanding set and outputs pk_e; otherwise, it outputs $\perp$.

GenURID(pk_e). The SP then computes the MRID using a cryptographic hash of the public key pk_e:

$$\text{mrid} = \text{SHA256}(pk_e).$$

4.2 URID Image Generation

After obtaining the MRID, the service transforms this machine-recognizable identifier into a user-recognizable image, satisfying the *user recognizable* usability requirement (Section 3). While we use images in this work, the same procedure can produce short animations, symbols, or text snippets. All URID generation occurs on the server.

Prompt Construction. Upon computing the MRID, we transform it deterministically into a text prompt for image generation. The MRID is used as a seed for Python's built-in `random` library. For each of the template's 8 slots (e.g., *object*, *background*, *style*, *lighting*), we select one option from a slot-specific set (ranging from 5 to 200 options). We then extract the lower 64 bits of the MRID as the seed for

the image generator's internal randomness. With this scheme, the template yields approximately 2.85 trillion distinct prompts. Even if prompts sharing one slot are considered too visually similar, the remaining combinations still yield more than 10^{12} distinct images.

Image Generation. We use the pre-trained Stable Diffusion XL Turbo (SDXL-Turbo) model [51], which offers high fidelity at low latency. Given the prompt and seed, we seed both Python and PyTorch random number generators to ensure perfect reproducibility without caching. SDXL-Turbo requires only 2–4 diffusion steps, enabling generation of roughly 10 images per second on a modern GPU after warm-up. The resulting image is returned to the service for inclusion in the ALH or login notifications.

Image-Generation Properties. This image-generation process is designed to satisfy three goals:

(1) **User distinguishability:** Different MRIDs should yield visually distinct images, minimizing the chance that a user mistakenly recognizes an attacker-generated URID.
(2) **Cryptographic binding:** Because the MRID is a SHA-256 hash of pk_e, the prompt and image are deterministically bound to the verified identity-specific device key.
(3) **Fast generation:** Generation must be fast enough for online login flows; SDXL-Turbo satisfies this requirement.

To evaluate property (1), we embed each generated image using CLIP [43] and analyze pairwise Euclidean distances in its learned visual-semantic feature space. This is a model-based proxy for similarity, not a direct measurement of human judgments. We generated 100,000 URID images using random MRIDs and computed all pairwise distances. For comparison, we computed the same metric on a random subset of 40,000 images from COCO Train 2017 [33]. The median CLIP distance among URID images was approximately 8.5, compared to 10.5 in COCO. Although URID images have slightly less variability (as expected from using a controlled template), even the closest pairs remained visually distinct when examined manually (see Fig. 5 in Appendix A). These measurements show no exact duplicate embeddings in our sample, but they do not establish a probability of human-perceptual collision.

Justification from Visual Memory Literature. Our design is informed by long-term visual recognition research. Mandler and Ritchey [35] studied memory for changes to pictures after repeated exposure and retention delays. Brady et al. [15] found high recognition accuracy for previously viewed objects, including an 87% result when a studied object was paired with the same object in a different state or pose. These findings establish that visual long-term memory can retain substantial object detail, but they do not show that seed-induced differences between generated images will always be perceptible. Our CLIP analysis and user study therefore evaluate complementary aspects of the generated identifiers.

These results characterize diversity in CLIP space; the user study provides the direct evidence about recognition in our task. Deliberate perceptual imitation is considered below.

Computational and Storage Overhead. The overhead of URIDs is small and is incurred at well-defined points. On the device, keypair generation inside the TEE happens once per identity (after a factory reset), and the Privacy CA is contacted only at initialization and for periodic certificate refresh; thereafter, each GetSignature request costs a single Keystore signing operation. On the SP side, processing a response requires one certificate-chain check, one signature verification, and one SHA-256 computation per login, and the resulting MRID is a single 32-byte value stored alongside the account. Image generation runs entirely on the server: SDXL-Turbo requires only 2–4 diffusion steps, and because the MRID-to-image mapping is deterministic, an SP can either cache each image once or regenerate it on demand without storing any additional state. None of these costs are on the critical path of authentication itself; the URID can be generated or fetched asynchronously after login succeeds.

4.3 Security and Privacy of URID

We formalize the guarantees from Section 3; complete experiments and proofs appear in Appendix C. We consider probabilistic polynomial-time (PPT) adversaries that control their own devices and protocol inputs and observe valid responses. The trusted components are those in Section 3.1. We assume that the device and CA signatures are *existentially unforgeable under chosen-message attack* (EUF-CMA): even after obtaining signatures on adaptively chosen messages, an adversary cannot produce a valid signature on a new message [38]. We model SHA-256 as second-preimage resistant — given x, finding $x' \neq x$ with the same hash is infeasible — and collision resistant — finding any distinct x, x' with the same hash is infeasible [38]. We further assume independently generated identity keys and Privacy CA certificates whose SP-visible distributions contain no device-specific information.

Security Definitions. Target-MRID unforgeability requires that an attacker cannot make an SP accept the existing MRID of an honest device without an authorized signature by that device. An attacker may present its own certified key for the victim's identity, but this produces a new MRID and hence an unfamiliar URID. *Freshness* requires that a response accepted for one challenge cannot be accepted for another or after that challenge is consumed. *MRID uniqueness* requires independently generated keys to produce equal MRIDs only with negligible probability. *Cross-identity unlinkability* requires that, for distinct identities, an SP cannot distinguish whether their protocol transcripts came from one device or two. We state the key security theorems below, each with a brief proof sketch; detailed proofs appear in Appendix C.

Theorem 4.1 (Authenticity and freshness). *Under the assumptions above, the advantage of any PPT adversary in the target-MRID unforgeability or freshness experiment is negligible.*

Proof sketch. If the SP accepts a target MRID, then either the adversary forged an authorized signature on the fresh tuple, forged the CA certificate chain, or found a SHA-256 second preimage. Freshness follows because the signed message binds $(\mathrm{app}_{\mathrm{id}}, e, r, ts)$ and the SP removes the outstanding tuple after acceptance. Therefore a successful attack implies a break of EUF-CMA security or SHA-256 hardness.

Theorem 4.2 (MRID uniqueness). *The probability that two independently generated public keys produce equal MRIDs is negligible.*

Proof sketch. Two equal MRIDs imply either equal public keys or a SHA-256 collision. Independent generation makes the first event

negligible, and SHA-256 collision resistance makes the second event negligible as well.

Theorem 4.3 (Cross-identity unlinkability). *For distinct identities, the SP-visible cryptographic transcripts produced by one device and by two devices are computationally indistinguishable.*

Proof sketch. The two worlds differ only in whether the identities share a device, but both cases generate keys and certificates from the same distribution and sign fresh messages with the same encoding. Hence any SP-visible transcripts are distributed identically up to negligible statistical differences, so the adversary's distinguishing advantage is negligible.

Scope and Benefits. Authenticity and freshness prevent another device from copying a familiar MRID or replaying a response across identities, applications, or challenges. Unlinkability prevents SPs from using protocol artifacts to correlate different identities on one device, while the public MRID contains neither e nor a persistent hardware identifier. This guarantee assumes the Privacy CA and the SP do not collude. It also excludes side information such as accounts, IP addresses, or timing. The CA sees attestation during issuance but not e; different services having access to the same user identity e receive the same pk_e and are intentionally linkable — they can link user activity across services using the identity e anyway. Finally, these proofs concern the MRID, not perceptual similarity of URIDs: an attacker could use trial and error to find MRIDs that produce perceptually similar URIDs. We hope such attacks are expensive and that the Privacy CA will have some form of rate limiting in certificate issuance to deter such an attack.

5 Recognition Memory of URID Images

To measure the effectiveness of URIDs, it is necessary to compare them to existing device identifiers, such as the IP address and user-agent string, or to information derived from these, as shown in the ALH today. To do this, we conducted a within-subjects user study with 59 participants, recruited in two waves. With this study, we sought to evaluate whether using the URID enables users to recognize malicious devices. We hypothesized that *participants would be able to identify malicious devices in the ALH more accurately when URIDs are present than when they are not.* We measured accuracy using precision, recall, and the F_1 score.

5.1 Study Design

We conducted a three-week, within-subjects study using a custom website modeled on familiar social-media applications (Fig. 3; screenshots in Appendix D). During Weeks 1 and 2, participants encountered URIDs while completing simulated login and browsing tasks. In Week 3, they classified login records with and without URIDs. This controlled procedure was designed to measure recognition under repeated exposure; it does not prescribe how frequently a deployed system would display or test URIDs.

Participants and Pre-Study Survey. We recruited U.S. adults via Prolific [41]; 59 participants were included in the analysis across two recruitment waves. Before the study, participants reported the number and types of devices they used, their operating systems and models, and their email address. The study website dynamically

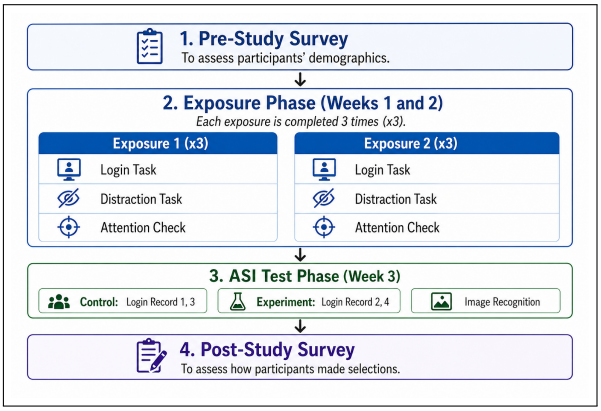


Figure 3: Overview of the experimental design. Participants completed two week-long learning phases followed by a testing phase containing simulated login scenarios with and without URIDs. The final recognition task evaluated participants' ability to recognize identifiers presented earlier in the study. Generated with ChatGPT image generation.

parsed browser and operating-system information from each user-agent string. Survey and parsed information were used to construct the simulated activity-log entries presented in Week 3. Participants accessed the website through a unique token sent by email. They received a total of $20 for completing the study. The protocol was approved by our university IRB as minimal-risk human-subjects research; further ethical considerations appear in the Ethical Considerations section.

Weeks 1–2: Exposure Tasks. Each learning week used a different set of images. In each session, participants logged into three simulated applications without entering a password. For each application, the interface displayed six URID images associated with legitimate devices for 20 seconds, after which participants completed a short distraction task, such as commenting on a fictitious post, ranking posts, or watching a video. The first wave completed three sessions per week, separated by at least 24 hours, so each identifier was encountered three times during its learning week. The second wave completed six sessions per week, producing six encounters per identifier. Across both weeks, participants encountered 36 distinct identifiers associated with six simulated applications. Eighteen identifiers were subsequently used in Week 3.

Participants were not told during the learning weeks that the images represented device identifiers or could later indicate whether a login was legitimate. The Week 3 target images were therefore encountered incidentally during the simulated login flow. The structured and repeated exposures were study instrumentation and may not reflect the frequency with which users would encounter URIDs in deployment.

Attention Checks. After each session, participants completed a four-option recognition check for an identifier they had just observed. They could select options until choosing the correct image, and we recorded the successful attempt number. These checks were

used to detect inattentive participation. Crucially, none of the images appearing as targets or distractors in an attention check was used in the Week 3 login-record tests. Participants thus received no recognition practice or corrective feedback for the identifiers on which login detection was evaluated. The checks may nevertheless have increased general attention to visual identifiers as a class.

Week 3: Login-Record Classification. Each participant completed four tests: two control tests showing conventional activity-log information and two experimental tests adding URIDs to the same kinds of information. Conventional fields included device type, operating system, browser, IP address, and location. Each test contained a six-record activity log drawn from the simulated applications used during learning. Participants selected every record they considered illegitimate without being told how many attacks were present. Each test contained one or two malicious records, with the remaining records representing legitimate prior activity. Test order and the positions of malicious records were randomized.

In the experimental tests, participants were told that a recognized image was likely associated with one of their earlier sessions. This instruction explained how the previously unexplained images could inform the classification task, but it also directed participants' attention to image familiarity. In practice such instructions can be added to the ALH interface.

Attack Conditions. The malicious records represented four combinations of whether the user agent (UA) and IP address matched metadata from the participant's simulated legitimate history:

- **Type 1: Unknown UA–Unknown IP.** Neither field matched prior legitimate activity, representing a login from an unfamiliar device and network.
- **Type 2: Known UA–Unknown IP.** The UA matched prior activity but the IP address did not, representing credential reuse in which the attacker copies the UA while connecting from another network.
- **Type 3: Unknown UA–Known IP.** The IP address matched prior activity but the UA did not. This can represent an attacker sharing the victim's network, using an endpoint on that network, or otherwise producing a login attributed to the same public IP.
- **Type 4: Known UA–Known IP.** Both fields matched prior activity. In the experimental interface, the unfamiliar URID was therefore the only displayed field distinguishing the malicious record from matching legitimate records. This condition models the interpersonal-attack scenario motivating URIDs, in which an attacker can access the same network and use a similar device.

Each participant was assigned two attack types, one for each control–experimental test pair. For unknown fields, the generated IP and location could range from a nearby city to another continent, while an unknown UA could differ in device, operating system, or browser. This variation increases ecological breadth but also produces variation in attack salience across participants.

Protocol Deviation. A programming error affected 21 of the 42 first-wave participants: each experimental test contained two malicious records and each control test contained one, instead of independently drawing one or two attacks for each test. The error was corrected for the remaining first-wave participants and did not affect the second wave. It created unequal attack prevalence between conditions and can therefore influence participants' response behavior as well as aggregate precision and F_1; normalized metrics do not by themselves remove this potential bias. We report the full-sample estimates together with a sensitivity analysis restricted to the 38 correctly configured participants in Section 6.

Delayed Image-Recognition Task. The final 21 first-wave participants and all 17 second-wave participants also completed an exploratory image-only recognition task, labeled Type 5. They classified a mixture of identifiers encountered during learning and new identifiers without contextual login information. The second wave used an inverted response format, so we analyze the task descriptively and do not treat the two implementations as directly equivalent. Results appear in Section 6.3.

5.2 Statistical Analysis Framework

Each login-record decision was classified as a true positive (a malicious record flagged), true negative (a legitimate record not flagged), false positive (a legitimate record flagged), or false negative (a malicious record not flagged). From these counts, we computed precision, recall, F_1, and accuracy separately for each participant and condition. Precision is the proportion of flagged records that were malicious, recall is the proportion of malicious records flagged, F_1 is the harmonic mean of precision and recall, and accuracy is the proportion of all records classified correctly. When a participant flagged no records, precision was undefined mathematically but set to zero by convention; because every test included at least one attack, recall was then zero, and F_1 was also set to zero.

Our inferential analyses use participant-level metrics, giving each participant equal weight. We also report pooled confusion matrices and metrics computed after summing decisions across participants. These pooled values are descriptive: they weight participants by their numbers of legitimate and malicious records and are affected by the condition-dependent prevalence error described in Section 5.1. They therefore need not equal the means of the participant-level metrics used for inference.

For each metric f and participant i, we calculated the within-participant difference

$$\Delta_i = f(\text{URID}, i) - f(\text{Control}, i).$$

We tested the directional hypothesis $H_1 : \mu_\Delta > 0$ against $H_0 : \mu_\Delta \leq 0$ using a one-sided paired-samples t-test. The direction was specified before analysis because the study asks whether adding a learned visual cue improves classification. We report the mean difference, its two-sided 95% confidence interval, the test statistic, and Cohen's d_z. A two-sided p-value cannot always be obtained by simply doubling the reported one-sided value; for the symmetric t distribution it is $2 \cdot \min(p, 1 - p)$.

We treat overall F_1 and Type 4 F_1 as the two primary comparisons: overall F_1 summarizes detection performance, while Type 4 tests the motivating case in which both conventional fields match prior legitimate activity. We control family-wise error across these two tests using the Holm–Bonferroni procedure and report adjusted p-values. Precision, recall, accuracy, and the remaining attack-specific comparisons are exploratory; their p-values are unadjusted

and are interpreted together with effect estimates and confidence intervals rather than as independent confirmatory findings. All tests use $\alpha = 0.05$.

Because participant-level differences are bounded and contain many zeros, we checked the two primary F_1 results with one-sided Wilcoxon signed-rank tests. These sensitivity tests led to the same threshold-based conclusions as the paired t-tests (overall F_1: $p = 0.034$; Type 4 F_1: $p = 0.009$). Attack-type analyses include only participants assigned the corresponding type and therefore have smaller, varying sample sizes.

The two recruitment waves differed in exposure frequency. To assess whether the estimated URID effect varied by wave, we compared the participant-level Δ_i values using a two-sample t-test, equivalent to testing a condition-by-wave interaction in this two-period summary. A nonsignificant interaction does not establish equivalence, so Section 6.2 reports the interaction estimate and confidence interval together with each wave's effect size. Because the interaction estimates were imprecise, the pooled result is interpreted as the average effect across the three- and six-exposure protocols.

The delayed image-recognition task (Type 5) was analyzed descriptively because it was administered only to a subset of participants and used different response formats across waves. We do not use it for inferential estimates of memory decay.

Results by attack type appear in Section 6.4.

6 Results

6.1 Participants and Overall Performance

In the first wave, 126 Prolific workers completed the recruitment survey, 82 began the study, 61 completed Week 1, 55 completed Week 2, and 50 completed all three phases. We excluded eight completers under the study's low-engagement criteria, leaving 42 participants for analysis. In the second wave, 53 workers completed recruitment, 24 began, 19 completed both learning weeks, and 17 completed testing; all 17 were retained. The primary login-record task used three exposures per learning week in the first wave and six in the second (Section 5.1). The supplementary Type 5 task also used a different response format in the second wave. We pool the $n = 59$ participants to estimate the average URID effect across the two exposure protocols and report wave-specific estimates in Section 6.2.

All 59 participants classified 1,416 simulated login records, 708 per condition. Mean participant-level F_1 was 0.581 with URIDs and 0.509 without them, a difference of +0.072 (95% CI $[-0.013, 0.157]$; $t(58) = 1.70$; one-sided Holm-adjusted $p = 0.048$; $d_z = 0.22$). Here and below, 58 denotes the paired-test degrees of freedom ($n-1$), not the number of participants. The Wilcoxon sensitivity test yielded $p = 0.034$. Thus, the primary directional test crossed the prespecified 0.05 threshold, but the effect was small and its two-sided confidence interval included zero. Exploratory differences in precision and recall were positive but uncertain (Table 1).

Because each six-entry test contained only one or two attacks, accuracy is dominated by legitimate entries: flagging nothing yields accuracy of 0.83 or 0.67. We therefore emphasize precision (the fraction of flags that are attacks), recall (the fraction of attacks detected), and their harmonic mean, F_1, while reporting accuracy for completeness.

Table 1: Participant-level paired comparisons ($n = 59$). Values in the p column are unadjusted and one-sided; the Holm-adjusted value for the primary overall-F_1 comparison is 0.048.

Metric	URID	Control	Mean Δ	t (df = 58)	p
F_1 Score	0.581	0.509	+0.072	1.70	0.048
Precision	0.580	0.509	+0.070	1.51	0.069
Recall	0.606	0.542	+0.064	1.48	0.072
Accuracy	0.845	0.845	0.000	0.00	0.500

Table 2: Pooled confusion matrices for the control and URID conditions. These descriptive counts are affected by unequal attack prevalence.

Condition	Malicious?	User Selection	
		True	False
Control	**True**	88 (TP)	74 (FN)
	False	36 (FP)	510 (TN)
URID	**True**	116 (TP)	76 (FN)
	False	34 (FP)	482 (TN)

Table 3: Pooled performance across conditions. These micro-averaged values are descriptive and differ from the participant-level means used for inference. Acc = accuracy, Pcn = precision, Rcl = recall.

Condition	# records	Acc	Pcn	Rcl	F_1
Control	708	0.845	0.710	0.543	0.615
URID	708	0.845	0.773	0.604	0.678

The pooled counts in Table 2 summarize all records but do not give each participant equal weight.

The conditions contain different numbers of attacks (162 control; 192 URID), largely because of the early-deployment error described in Section 5.1; raw counts are therefore not directly comparable. Pooled recall was 0.543 (88/162) in control and 0.604 (116/192) with URIDs, while the false-positive rate was 0.066 in both conditions. Pooled precision was 0.710 versus 0.773 and F_1 was 0.615 versus 0.678 (Table 3). These descriptive pooled metrics are affected by the prevalence imbalance; the inferential tests use paired, per-participant values (Table 1), with a sensitivity analysis excluding affected participants below.

At the participant level, exploratory precision (+0.070, $t(58) = 1.51$, unadjusted $p = 0.069$) and recall (+0.064, $t(58) = 1.48$, unadjusted $p = 0.072$) estimates favored URIDs, while mean accuracy was identical. These results do not establish effects on the individual component metrics.

Robustness to the Configuration Error. Restricting the analysis to the 38 correctly configured participants produced similar point estimates for overall F_1 (+0.069, $t(37) = 1.15$, unadjusted $p = 0.13$), Type 4 F_1 (+0.125, $t(20) = 1.56$, unadjusted $p = 0.07$), and recall

Table 4: Descriptive performance by wave and condition. Acc = accuracy, Pcn = precision, Rcl = recall.

Wave	Cond.	# records	Acc	Pcn	Rcl	F_1
Wave 1	Control	504	0.853	0.711	0.541	0.615
(3 exposures, $n = 42$)	URID	504	0.839	0.787	0.594	0.677
Wave 2	Control	204	0.824	0.707	0.547	0.617
(6 exposures, $n = 17$)	URID	204	0.858	0.738	0.633	0.681

(+0.079, unadjusted $p = 0.09$), but with wide uncertainty. Differences between the affected and unaffected groups were also imprecise (Welch tests: overall F_1, $p = 0.90$; Type 4 F_1, $p = 0.58$). The restricted analysis therefore does not reveal a clear reversal, but neither does it rule out bias from the configuration error.

6.2 Per-Wave Results

For the primary login-record task, each URID was shown three times per learning week in the first wave and six times in the second (Section 5.1). Table 4 reports pooled descriptive metrics by wave, while Table 5 reports participant-level paired effects.

Within each wave, point estimates for F_1, precision, and recall favored URIDs, but neither wave provided a precise stand-alone estimate. The overall-F_1 interaction was -0.02 for the six- versus three-exposure wave (95% CI $[-0.21, 0.17]$; $t(57) = -0.21$; two-sided $p = 0.83$). For Type 4 F_1, it was -0.11 (95% CI $[-0.40, 0.18]$; $t(30) = -0.77$; two-sided $p = 0.45$). These intervals are wide relative to the pooled F_1 effect of +0.072. The data therefore do not resolve whether exposure frequency changes the effect; pooling estimates the average across both protocols.

Neither wave was individually powered for the small pooled effect ($d_z = 0.22$): at 80% power and one-sided $\alpha = 0.05$, the detectable effects were approximately $d_z = 0.39$ for the first wave and $d_z = 0.63$ for the second. These post hoc calculations describe the limited precision of the subgroup estimates rather than evidence for equivalence.

6.3 Exploratory Delayed Image Recognition (Type 5)

The final 21 participants in the three-exposure wave and all 17 participants in the six-exposure wave completed an image-only task containing previously seen and unseen images. The six-exposure wave used an inverted response format. Because administration was not uniform across the sample, we treat these results as exploratory.

The immediate attention check and delayed task used different images and response formats and are not repeated measurements of the same items. For descriptive comparison, the immediate four-choice task assigned scores of 100, 66.7, 33.3, or 0 for a correct selection on attempts one through four: score $= 100(4 - a)/3$. The delayed score was the percentage of previously seen images classified as familiar, calculated separately for Week 2 images (one-week delay) and Week 1 images (two-week delay).

In the three-exposure wave, mean scores were 91.2 on the immediate task, 84.7 for Week 2 images, and 81.0 for Week 1 images.

Table 5: Paired URID-minus-control effects by wave. Mean Δ with two-sided 95% CI, Cohen's d_z, and unadjusted one-sided p. The first and second waves used three and six exposures per identifier per learning week, respectively (3 exp. and 6 exp.). Overall and Type 4 F_1 are primary in the pooled analysis; wave-specific comparisons are descriptive.

Metric	Wave	n	Mean Δ [95% CI]	d_z	p
Overall F_1	3 exp.	42	+0.078 $[-0.033, 0.189]$	0.22	0.082
	6 exp.	17	+0.058 $[-0.070, 0.186]$	0.23	0.175
	pooled	59	+0.072 $[-0.013, 0.157]$	0.22	0.047
Precision	3 exp.	42	+0.088 $[-0.027, 0.203]$	0.24	0.065
	6 exp.	17	+0.026 $[-0.148, 0.201]$	0.08	0.376
	pooled	59	+0.070 $[-0.023, 0.164]$	0.20	0.069
Recall	3 exp.	42	+0.054 $[-0.061, 0.168]$	0.15	0.176
	6 exp.	17	+0.088 $[-0.023, 0.199]$	0.41	0.055
	pooled	59	+0.064 $[-0.022, 0.150]$	0.19	0.072
Type 4 F_1	3 exp.	21	+0.192 $[0.001, 0.383]$	0.46	0.024
	6 exp.	11	+0.082 $[-0.119, 0.282]$	0.27	0.192
	pooled	32	+0.154 $[0.017, 0.292]$	0.40	0.015

Table 6: Pooled confusion-matrix counts by attack type. Raw counts are descriptive because attack prevalence differed between conditions.

Attack Type	Condition	TP	TN	FP	FN
Type 1	Control	39	156	7	8
(Unknown UA–Unknown IP)	URID	48	151	4	7
Type 2	Control	30	125	9	10
(Known UA–Unknown IP)	URID	39	120	7	8
Type 3	Control	14	86	12	20
(Unknown UA–Known IP)	URID	14	89	9	20
Type 4	Control	5	143	8	36
(Known UA–Known IP)	URID	15	122	14	41

In the six-exposure wave, the immediate score was 92.9 and the Week 1-image score was 75.2; this wave did not test Week 2 images and reversed which response participants selected.

These values are familiar-image hit rates rather than complete measures of recognition: they do not incorporate false alarms to unseen images. Consequently, they cannot establish a fixed chance baseline, discrimination ability, or a rate of memory decay. They show only that participants continued to label many previously seen images as familiar after one or two weeks. A separate follow-up study of a single ambiently displayed identifier is described in Appendix B.

6.4 Effect by Attack Type

We next report exploratory participant-level results for the four UA–IP combinations defined in Section 5.1. These scenarios assume that malicious sessions remain visible in the activity log; attacks that suppress or alter the log are outside the evaluation. Pooled counts by attack type appear in Table 6, and participant-level comparisons appear in Table 7.

Table 7: Participant-level comparisons by attack type. Δ denotes the mean difference; positive differences favor URIDs. Values in the p column are unadjusted and one-sided; the Holm-adjusted value for the primary Type 4 F_1 comparison is 0.029. All other rows are exploratory.

Attack Type / Metric	n	URID	Control	Δ	t (df)	p
Type 1						
F_1 Score	35	0.839	0.781	+0.058	0.76 (34)	0.227
Precision	35	0.830	0.776	+0.054	0.66 (34)	0.258
Recall	35	0.857	0.814	+0.043	0.59 (34)	0.278
Accuracy	35	0.948	0.929	+0.019	0.73 (34)	0.237
Type 2						
F_1 Score	29	0.791	0.717	+0.074	0.83 (28)	0.206
Precision	29	0.787	0.707	+0.080	0.86 (28)	0.198
Recall	29	0.828	0.759	+0.069	0.78 (28)	0.221
Accuracy	29	0.914	0.891	+0.023	0.68 (28)	0.251
Type 3						
F_1 Score	22	0.377	0.405	−0.027	−0.22 (21)	0.587
Precision	22	0.348	0.436	−0.087	−0.67 (21)	0.746
Recall	22	0.432	0.432	0.000	0.00 (21)	0.500
Accuracy	22	0.780	0.758	+0.023	0.33 (21)	0.371
Type 4						
F_1 Score	32	0.248	0.094	+0.154	2.29 (31)	0.015
Precision	32	0.276	0.089	+0.187	2.35 (31)	0.013
Recall	32	0.250	0.125	+0.125	1.68 (31)	0.052
Accuracy	32	0.714	0.771	−0.057	−2.00 (31)	0.973

Type 1: Unknown UA–Unknown IP. Both conventional fields differed from prior activity. Mean F_1 was 0.839 with URIDs and 0.781 in control ($\Delta = +0.058$; $t(34) = 0.76$; unadjusted $p = 0.227$). Precision and recall estimates were also positive but imprecise (Table 7). High control performance suggests that the conventional mismatches already made these records conspicuous.

Type 2: Known UA–Unknown IP. The UA matched prior activity but the IP did not. Mean F_1 was 0.791 with URIDs and 0.717 in control ($\Delta = +0.074$; $t(28) = 0.83$; unadjusted $p = 0.206$). Precision and recall estimates also favored URIDs, but none was precise enough to distinguish from zero.

Type 3: Unknown UA–Known IP. The IP matched prior activity but the UA did not. Mean F_1 was 0.377 with URIDs and 0.405 in control ($\Delta = -0.027$; $t(21) = -0.22$; unadjusted $p = 0.587$); recall was identical and precision was lower with URIDs. These estimates provide no evidence of a benefit. Although survey respondents frequently mentioned location, the study did not manipulate attention and cannot establish that location cues caused this result.

Type 4: Known UA–Known IP. Both conventional fields matched prior activity; in the experimental interface, only the unfamiliar URID distinguished a malicious entry from matching legitimate entries. Raw counts should not be compared directly because attack prevalence differed between conditions.

Mean F_1 was 0.248 with URIDs and 0.094 in control ($\Delta = +0.154$; 95% CI $[0.017, 0.292]$; $t(31) = 2.29$; Holm-adjusted one-sided $p = 0.029$; $d_z = 0.40$). The Wilcoxon sensitivity test gave $p = 0.009$. Exploratory precision was also higher ($\Delta = +0.187$; unadjusted $p = 0.013$), whereas the recall estimate was uncertain ($\Delta = +0.125$; unadjusted $p = 0.052$). This was the largest observed URID effect, but the correctly configured subset estimate was less precise ($\Delta = +0.125$; $p = 0.07$).

Mean accuracy decreased by 0.057 (two-sided $p = 0.054$), reflecting additional errors on the more numerous legitimate records. Although accuracy is dominated by the majority class, this decrease represents a potentially important false-alarm cost. The study supports a recognition benefit under this simulated condition; whether deployed URIDs resist spoofing is a separate system-security question.

6.5 Post-Study Survey

Fifty-four of 59 participants completed the optional post-study questionnaire. Responses were linked using access codes or, when that field contained another identifier, recorded IP addresses; for duplicate submissions, we retained the first complete response. Of these respondents, 67% described themselves as "Very familiar" or "Expert" with technology and 61% reported previously checking login activity (Table 8). These self-reports characterize the respondent subset and may not generalize to all participants.

Forty-six of the 54 respondents (85%) recalled noticing the AI-generated images. This shows reported awareness among survey respondents, but does not establish accurate recognition or explain the behavioral results.

Participants also answered: *"How did you decide whether a login record was legitimate or illegitimate?"* Following thematic analysis [16], two researchers independently coded all responses and resolved disagreements through discussion. Location and device name were the dominant reported heuristics. For example:

> *"I mostly looked at the **city or IP location**. If it wasn't where I usually log in from, I marked it as suspicious."* (P12)

Another participant cross-checked device and location:

> *"I compared the **device name and location**. If both matched, I thought it was safe."* (P5)

More than two-thirds of respondents mentioned geographic cues, whereas fewer than one-third mentioned either the AI-generated image or device model. This pattern is consistent with participants relying on familiar activity-log fields, although the questionnaire cannot establish which cues caused their classifications.

No participant reported using URIDs as their primary cue, and several expressed uncertainty about remembering them:

> *"Sometimes I wasn't sure, so I just used the **city** as my main clue because the images all looked similar after a while."* (P27)

Despite these reports, mean participant-level F_1 was higher with URIDs. The discrepancy is compatible with, but does not demonstrate, an implicit-familiarity mechanism [27, 52]. The imprecise wave comparison likewise cannot determine whether doubling exposure changes detection performance (Section 6.2).

Table 8: Post-study survey responses ($n = 54$). Each entry reports count (percentage).

Question	Response	n (%)	Question	Response	n (%)
Age	18–29	12 (22.2%)	Tech familiarity	Slightly	4 (7.4%)
	30–44	19 (35.2%)		Moderately	14 (25.9%)
	45–60	18 (33.3%)		Very	31 (57.4%)
	60 or older	5 (9.3%)		Expert	5 (9.3%)
Gender	Male	32 (59.3%)	Checked login activity	Yes	33 (61.1%)
	Female	22 (40.7%)		Maybe	10 (18.5%)
Education	High school or less	7 (13.0%)		No	11 (20.4%)
	Some college/associate	16 (29.6%)	Noticed AI images	Yes	46 (85.2%)
	Bachelor's	22 (40.7%)		No	6 (11.1%)
	Graduate/professional	9 (16.7%)		Not sure	2 (3.7%)

7 Discussion

In our study, AI-generated visual device identifiers (URIDs) were associated with a modest but statistically significant improvement in login-detection performance. The overall F_1 increase was significant ($t(58) = 1.70$, Holm-adjusted $p = 0.048$, $d_z = 0.22$), with precision and recall moving in the same direction. Participants also achieved higher pooled precision with URIDs than in the control condition (0.77 vs. 0.71) without more false alarms. The strongest effect occurred in the most deceptive condition (Type 4), where both UA and IP matched legitimate sessions: F_1 was higher with URIDs ($t(31) = 2.29$, Holm-adjusted $p = 0.029$, $d_z = 0.40$), as was precision in an exploratory comparison (uncorrected $p = 0.013$, $d_z = 0.42$). These results suggest that a stable visual cue can reveal otherwise benign-looking attacks, but evidence from a controlled study does not guarantee real-world effectiveness.

These gains followed only three exposures per image in the first wave and six in the second. We found no evidence that doubling exposure changed the effect (Section 6.2), although the second wave was too small to rule out moderate differences, so we cannot say whether recognition saturates after a few encounters. Participants learned 36 distinct URIDs, of which a random 18 appeared during testing — likely more than most users would need to recognize. Users with many devices or accounts would encounter their identifiers frequently, while those with fewer devices would have fewer to learn. Recognition may therefore strengthen through natural exposure after deployment. However, other study elements may have inflated the benefit, as discussed below, so we do not treat the observed effects as a lower bound on real-world performance.

Limitations. Our study used simulated interfaces that may not capture real-world distractions, time pressure, or competing demands on attention. Participants reviewed logs in a controlled setting, and the fixed two-week interval may not reflect how often users inspect account histories. The sample also skewed young and technologically literate, limiting generalizability.

The study design may also have affected the results. Post-login attention checks ensured engagement but may have reinforced memorization beyond passive exposure. Participants were also told that recognized images indicated legitimate sessions, potentially priming them to attend to familiarity; deployed systems would instead teach the role of URIDs through onboarding or gradual experience. These effects cut in both directions: participants saw each identifier less often than they might in practice, potentially understating recognition, while the instructions and attention checks may have overstated attention to the images. Our estimates therefore indicate an effect but do not bound deployed performance.

Figure 4: Different places where a URID can be embedded so that the user becomes familiar with the image without active training. These illustrative mock-ups were generated with ChatGPT image generation.

Future work should evaluate more diverse populations, naturalistic alert settings, and longer retention intervals.

Value of URIDs Under Contextual Ambiguity. URIDs provided the greatest benefit when contextual cues failed. In Type 4, both IP and UA matched prior sessions, leaving traditional indicators with no signal of compromise, yet URIDs enabled significantly better discrimination. Visual identifiers can thus serve as a cognitive backstop: a familiar cue that remains difficult to spoof even when an attacker replicates device and network metadata. Such ambiguity arises in high-capability threats, including insider attacks and interpersonal abuse, where adversaries may have physical proximity, contextual knowledge, or partial device access.

Implications for Real-World Design. URIDs should complement, not replace, contextual fields such as IP or location. Their value is an additional, low-cognitive-load signal that remains stable across sessions and resists spoofing. Because recognition is largely implicit, interfaces should build familiarity unobtrusively through login screens, loading animations, or account dashboards (Fig. 4). An SP would (i) request the MRID after a successful login, (ii) render its image beside ALH entries and in login notifications, and (iii) display the current device's image ambiently in the user interface. Although showing an identifier to an authenticated user may seem redundant, repeated exposure during routine use allows an unfamiliar image to stand out later without explicit training. Because each device–identity pair has one image, even a long log contains only as many distinct images as the user has devices; users can group entries by image and, where failed attempts are logged, focus on successful logins.

For high-risk users, particularly those facing intimate partner violence (IPV), URIDs may provide a lightweight defense against contextual spoofing. An abuser in the same household may share the victim's IP address and location and use a similar device model, making conventional cues uninformative, while the attacker's device still carries a distinct URID. Direct use of the victim's device, however, produces the familiar identifier and receives no additional protection; this case is outside our threat model. IPV deployments should also provide clear onboarding, opt-outs, and appropriate safety measures to avoid increasing risk.

Alternative Visual Representations and Scope of the Usability Claim. Our study shows that a stable, hard-to-spoof visual identifier improves detection; it does not establish that recognition requires TEE-attested, AI-generated images. A stable non-attested image — such as a user-chosen photo, emoji, or hash-based identicon — might provide similar recognition because the image only renders the underlying identifier. Representation and attestation, however, solve different problems. An identifier derived from a cookie or browser fingerprint can be cloned, placing the victim's familiar image on an attacker's session and creating false assurance. User-chosen photos and emoji are also forgeable, non-unique, and potentially privacy-sensitive, violating Section 3. Identicons derived from an attested MRID remain a viable alternative. We chose object-centric AI-generated images because recognition-memory research suggests that they support distinctive, durable recognition. Thus, our study establishes the usability benefit of stable visual identifiers, while hardware attestation prevents spoofing of the identifier beneath them.

Log-Based Grouping as a Complementary Approach. Rather than rely on prior recognition, an ALH could use visual identifiers to group entries by device. A user might inspect every login carrying their laptop's "elephant" image, verify its timestamps, and then examine the next group. This strategy avoids dependence on recognition memory and can expose misuse of a stolen device: although the device retains its familiar identifier, anomalous timestamps within its group may reveal the attack. The tradeoff is greater cognitive effort from systematically inspecting and cross-referencing entries. URIDs also support real-time notifications, where a quick familiarity judgment is more practical. The approaches are therefore complementary: grouping supports detailed investigation, while recognition supports lightweight, in-the-moment verification.

Client-Side Versus Server-Side Image Generation. Our design generates images on the server. Services using different models or configurations may therefore render the same MRID differently. Because URIDs are already scoped to a device–identity pair, users can learn service-specific images without violating the unlinkability goal. Client-side generation could instead provide a consistent image across services sharing an identity, but adds computation, model distribution, and version-management costs. A middle ground is to standardize the model, prompt template, and seeding protocol so that conforming servers produce identical outputs. We leave these tradeoffs to future work.

Memory Retention and Future Work. Participants retained URIDs well: recognition scores were 91.2 immediately after exposure, 84.7 after one week, and 81.0 after two weeks in the three-exposure wave, with comparable retention in the six-exposure wave. Such persistence may support detection even when users review login records infrequently. Future work should conduct longitudinal field studies, adapt URIDs to users' device ecosystems, and integrate them with passive defenses such as device fingerprints or behavioral biometrics. Combining these layers may strengthen protection with little user burden.

8 Conclusion

ALH interfaces are intended to help users spot suspicious logins, yet their usefulness is limited by imprecise, easily spoofed information and interfaces that are difficult to interpret. We introduced a new class of device identifiers that are *unspoofable*, *unlinkable*, and *user-recognizable*. Our approach derives stable, privacy-preserving identifiers from trusted hardware on modern phones and converts them into distinctive AI-generated images (URIDs) to support human recognition.

We showed that these identifiers can be generated using standard attestation mechanisms, and we designed them to integrate securely into existing login-notification workflows. In a study with 59 participants, URIDs were associated with improved login discrimination after brief exposure, with the largest directional gains in the most deceptive scenario (Type 4), where conventional textual and contextual cues fail. This pattern suggests that a stable visual cue can help users surface attacks that otherwise appear legitimate, but the effect should be interpreted in the context of a controlled user study rather than as a guarantee of real-world performance.

Participants' strong reliance on IP-based location limited benefits in network-spoofing attacks, underscoring that URIDs should complement — not replace — contextual indicators and that ALH interfaces must account for users' existing mental models. Longer-term exposure, field deployments, and evaluations in higher-risk contexts remain important avenues for future work.

Overall, URIDs show how privacy-preserving, user-recognizable device identifiers can enhance the ALH by aligning cryptographic guarantees with human memory and visual recognition.

Acknowledgments

We sincerely thank the anonymous reviewers and our shepherd for their insightful comments, which greatly improved this work. We acknowledge the generous funding support from NSF award #2339679 and the Baldwin Wisconsin Idea Endowment.

Ethical Considerations

Our study did not involve IPV survivors; rather, we recruited general users because our approach applies broadly to any account-holder. We designed all study procedures to minimize privacy risks and avoid teaching unsafe digital behaviors.

We collected participants' email addresses solely to send reminders for subsequent study phases. All links sent via email were presented in full plain text (i.e., without embedded HTML hyperlinks) to avoid encouraging risky click-through habits. For the same reason, participants were never asked to create or enter passwords. Instead, each participant was assigned a randomly generated pseudonymous username, used only within the study interface.

The study system stored only the pseudonymous username, the generated identifiers, and test results in an EC2-hosted database. No identifying information was kept on this server. Contact information (email addresses) was stored separately on an isolated, access-controlled machine within our laboratory. This separation ensured that no single system contained both identity and study data.

Finally, we submitted our protocol to our university's Institutional Review Board (IRB). The study was approved as minimal-risk human-subjects research.

Open Science

Our artifacts include:

- The Android application source code for MRID generation and TEE-based attestation.
- The web-based study interface used for participant interaction during the three-week user study.
- The URID image generation pipeline, including the prompt template and SDXL-Turbo configuration.

All artifacts are available for review at the following anonymous repository: https://anonymous.4open.science/r/codebase-8751/.

AI Use

The authors used generative AI-based tools, specifically ChatGPT and Claude, to revise the text, improve flow, and correct any typos, grammatical errors, and awkward phrasing. Figures 3 and 4 were generated with ChatGPT image generation. The Open Science artifacts were redacted with Claude Code to preserve anonymity and remove sensitive information. We have manually verified and are responsible for the accuracy, originality, and integrity of the output of all AI-based tools.

References

[1] Devdatta Akhawe and Adrienne Porter Felt. 2013. Alice in warningland: A large-scale field study of browser security warning effectiveness. In *22nd USENIX Security Symposium (USENIX Security 13)*. USENIX Association, Washington, DC, 257–272. https://www.usenix.org/conference/usenixsecurity13/technical-sessions/presentation/akhawe

[2] Android. 2024. Hardware-backed Keystore. https://source.android.com/docs/security/features/keystore. Accessed: 2024-10-31.

[3] Android. 2024. Hardware security module. https://developer.android.com/privacy-and-security/keystore#HardwareSecurityModule. Accessed: 2024-10-31.

[4] Android. 2024. Key and ID attestation. https://source.android.com/docs/security/features/keystore/attestation. Accessed: 2024-10-31.

[5] Android. 2024. Trusty TEE. https://source.android.com/docs/security/features/trusty. Accessed: 2024-10-31.

[6] Android Developers. 2019. Privacy changes in Android 10. https://developer.android.com/about/versions/10/privacy/changes. Accessed: 2023-07-22.

[7] Android1500. 2023. AndroidFaker releases. https://github.com/Android1500/AndroidFaker/releases. GitHub repository; accessed: 2023-07-22.

[8] Julio Angulo and Martin Ortlieb. 2015. "WTH..!?!" Experiences, reactions, and expectations related to online privacy panic situations. In *Eleventh Symposium on Usable Privacy and Security (SOUPS 2015)*. USENIX Association, Ottawa, Canada, 19–38. https://www.usenix.org/conference/soups2015/proceedings/presentation/angulo

[9] Apple Inc. 2021. *iCloud Private Relay Overview*. Technical Report. Apple Inc. https://www.apple.com/icloud/docs/iCloud_Private_Relay_Overview_Dec2021.pdf

[10] Apple Inc. 2023. Apple ID & Privacy. https://www.apple.com/legal/privacy/data/en/apple-id/. Accessed: 2023-07-22.

[11] Luca Arnaboldi, David Aspinall, Christina Kolb, and Saša Radomirović. 2023. Tactics for account access graphs. In *European Symposium on Research in Computer Security*. Springer, Springer, Cham, 452–470.

[12] Mozhgan Azimpourkivi, Umut Topkara, and Bogdan Carbunar. 2020. Human Distinguishable Visual Key Fingerprints. In *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, Virtual Conference, 2237–2254. https://www.usenix.org/conference/usenixsecurity20/presentation/azimpourkivi

[13] Mihir Bellare and Bennet Yee. 1997. *Forward integrity for secure audit logs*. Technical Report. Technical report, Computer Science and Engineering Department, University of ....

[14] Arkaprabha Bhattacharya, Alaa Daffalla, Kevin Lee, Rosanna Bellini, Nicola Dell, and Thomas Ristenpart. 2026. Inconsistent, Incomplete, and Insecure: A Survey of Account Security Interfaces. In *35th USENIX Security Symposium (USENIX Security 26)*. 5533–5552.

[15] Timothy F Brady, Talia Konkle, George A Alvarez, and Aude Oliva. 2008. Visual long-term memory has a massive storage capacity for object details. *Proceedings of the National Academy of Sciences* 105, 38 (2008), 14325–14329. doi:10.1073/pnas.0803390105

[16] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology* 3, 2 (2006), 77–101. doi:10.1191/1478088706qp063oa

[17] Raluca Budiu. 2024. Recognition vs. Recall in User Interfaces. https://www.nngroup.com/articles/recognition-and-recall/. Nielsen Norman Group.

[18] Cheun Ngen Chong, Zhonghong Peng, and Pieter H. Hartel. 2003. Secure Audit Logging with Tamper-Resistant Hardware. In *Security and Privacy in the Age of Uncertainty*, Dimitris Gritzalis, Sabrina De Capitani di Vimercati, Pierangela Samarati, and Sokratis Katsikas (Eds.). Springer US, Boston, MA, 73–84.

[19] Mary Czerwinski and Eric Horvitz. 2002. An investigation of memory for daily computing events. In *People and Computers XVI-Memorable Yet Invisible: Proceedings of HCI 2002*. Springer, Springer, London, UK, 229–245.

[20] Alaa Daffalla, Marina Bohuk, Nicola Dell, Rosanna Bellini, and Thomas Ristenpart. 2023. Account Security Interfaces: Important, Unintuitive, and Untrustworthy. In *32nd USENIX Security Symposium (USENIX Security 23)*. USENIX Association, Anaheim, CA, 3601–3618. https://www.usenix.org/conference/usenixsecurity23/presentation/daffalla

[21] Alaa Daffalla, Grace Myers, Rosanna Bellini, Thomas Ristenpart, and Nicola Dell. 2026. "Maybe there's only one passkey?": Challenges Investigating and Remediating Adversarial Passkeys. (2026).

[22] Emma Dauterman, Danny Lin, Henry Corrigan-Gibbs, and David Mazières. 2023. Accountable authentication with privacy protection: The Larch system for universal login. In *17th USENIX Symposium on Operating Systems Design and Implementation (OSDI 23)*. USENIX Association, Boston, MA, 81–98. https://www.usenix.org/conference/osdi23/presentation/dauterman

[23] Rachna Dhamija and J Doug Tygar. 2005. The battle against phishing: Dynamic security skins. In *Proceedings of the 2005 symposium on Usable privacy and security*. ACM, Association for Computing Machinery, New York, NY, USA, 77–88. doi:10.1145/1073001.1073009

[24] Paul Dunphy, Andreas P Heiner, and N Asokan. 2010. A closer look at recognition-based graphical passwords on mobile devices. In *Proceedings of the Sixth Symposium on Usable Privacy and Security*. Association for Computing Machinery, New York, NY, USA, 1–12. doi:10.1145/1837110.1837114

[25] Manuel Egele, Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. 2013. Compa: Detecting compromised accounts on social networks.. In *NDSS*, Vol. 13. Internet Society, Reston, VA, 83–91.

[26] Maximilian Golla, Miranda Wei, Juliette Hainline, Lydia Filipe, Markus Dürmuth, Elissa Redmiles, and Blase Ur. 2018. " What was that site doing with my Facebook password?" Designing Password-Reuse Notifications. In *Proceedings of the 2018*

ACM SIGSAC Conference on Computer and Communications Security. Association for Computing Machinery, New York, NY, USA, 1549–1566. doi:10.1145/3243734.3243767

[27] Peter Graf and Michael E. J. Masson (Eds.). 1993. *Implicit Memory: New Directions in Cognition, Development, and Neuropsychology*. Lawrence Erlbaum Associates, Hillsdale, NJ.

[28] Frank Haist, Arthur P. Shimamura, and Larry R. Squire. 1992. On the relationship between recall and recognition memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 18, 4 (1992), 691–702. doi:10.1037/0278-7393.18.4.691

[29] Sven Hammann, Saša Radomirović, Ralf Sasse, and David Basin. 2019. User account access graphs. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. Association for Computing Machinery, New York, NY, USA, 1405–1422. doi:10.1145/3319535.3354193

[30] Jason E Holt and Kent E Seamons. 2005. Logcrypt: forward security and public verification for secure audit logs. *Cryptology ePrint Archive* (2005).

[31] Jun Ho Huh, Hyoungshick Kim, Swathi SVP Rayala, Rakesh B Bobba, and Konstantin Beznosov. 2017. I'm too busy to reset my LinkedIn password: On the effectiveness of password reset emails. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 387–391.

[32] Vishal Karande, Erick Bauman, Zhiqiang Lin, and Latifur Khan. 2017. SGX-Log: Securing System Logs With SGX. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security* (Abu Dhabi, United Arab Emirates) *(ASIA CCS '17)*. Association for Computing Machinery, 19–30.

[33] Tsung-Yi Lin, Michael Maire, Serge Belongie, et al. 2014. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision (ECCV)*. Springer, Cham, 740–755.

[34] Di Ma and Gene Tsudik. 2009. A new approach to secure logging. *ACM Trans. Storage* 5, 1, Article 2 (March 2009), 21 pages.

[35] Jean M. Mandler and Gary H. Ritchey. 1977. Long-term Memory for Pictures: Effects of Repetition and Response Type. *Journal of Experimental Psychology: Human Learning and Memory* 3, 4 (1977), 386–396.

[36] Philipp Markert, Andrick Adhikari, and Sanchari Das. 2023. A Transcontinental Analysis of Account Remediation Protocols of Popular Websites. arXiv:2302.01401. https://arxiv.org/abs/2302.01401

[37] Philipp Markert, Leona Lassak, Maximilian Golla, and Markus Dürmuth. 2024. Understanding Users' Interaction with Login Notifications. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, Article 853, 17 pages. doi:10.1145/3613904.3642823

[38] Alfred J Menezes, Paul C Van Oorschot, and Scott A Vanstone. 1996. *Handbook of applied cryptography*. CRC press, Boca Raton, FL.

[39] Lorenzo Neil, Elijah Bouma-Sims, Evan Lafontaine, Yasemin Acar, and Bradley Reaves. 2021. Investigating Web Service Account Remediation Advice. In *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)*. USENIX Association, Virtual Conference, 359–376. https://www.usenix.org/conference/soups2021/presentation/neil

[40] Carolina Ortega Pérez and Alaa Daffalla. 2025. Encrypted Access Logging for Online Accounts: Device Attributions without Device Tracking. In *34th USENIX Security Symposium (USENIX Security 25)*. USENIX Association, Seattle, WA, 6679–6697. https://www.usenix.org/conference/usenixsecurity25/presentation/ortega-perez

[41] Stefan Palan and Christian Schitter. 2018. Prolific. ac—A subject pool for online experiments. *Journal of behavioral and experimental finance* 17 (2018), 22–27.

[42] Martin Pirker, Ronald Toegl, Daniel Hein, and Peter Danner. 2009. A PrivacyCA for Anonymity and Trust. In *Trusted Computing*, Liqun Chen, Chris J. Mitchell, and Andrew Martin (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 101–119.

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, et al. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*. PMLR, Virtual Conference, 8748–8763.

[44] Elissa M Redmiles. 2019. " Should I Worry?" A Cross-Cultural Examination of Account Security Incident Response. In *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, IEEE, San Francisco, CA, 920–934.

[45] Xin Ruan, Zhenyu Wu, Haining Wang, and Sushil Jajodia. 2015. Profiling online social behaviors for compromised account detection. *IEEE transactions on information forensics and security* 11, 1 (2015), 176–187. doi:10.1109/TIFS.2015.2482465

[46] Michael D. Rugg and Andrew P. Yonelinas. 2003. Human recognition memory: A cognitive neuroscience perspective. *Trends in Cognitive Sciences* 7, 7 (2003), 313–319. doi:10.1016/S1364-6613(03)00131-1

[47] Nicole C. Rust and Vahid Mehrpour. 2020. Understanding image memorability. *Trends in Cognitive Sciences* 24, 7 (2020), 557–568. doi:10.1016/j.tics.2020.04.001

[48] Mohamed Sabt and Jacques Traorè. 2016. Breaking into the KeyStore: A Practical Forgery Attack Against Android KeyStore. In *Computer Security – ESORICS 2016*, Ioannis Askoxylakis, Sotiris Ioannidis, Sokratis Katsikas, and Catherine Meadows (Eds.). Springer International Publishing, Cham, 531–548.

[49] Sena Sahin, Burak Sahin, and Frank Li. 2025. Was This You? Investigating the Design Considerations for Suspicious Login Notifications. In *Network and Distributed System Security Symposium (NDSS 2025)*. Internet Society, San Diego, CA, 15 pages. doi:10.14722/ndss.2025.240694

[50] Lynnmarie Sardinha, Mathieu Maheu-Giroux, Heidi Stöckl, Sarah Rachel Meyer, and Claudia García-Moreno. 2022. Global, regional, and national prevalence estimates of physical or sexual, or both, intimate partner violence against women in 2018. *The Lancet* 399, 10327 (2022), 803–813.

[51] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. 2023. Adversarial Diffusion Distillation. arXiv:2311.17042 [cs.CV] https://arxiv.org/abs/2311.17042 Model available at https://huggingface.co/stabilityai/sdxl-turbo.

[52] Daniel Schacter. 1987. Implicit Memory: History and Current Status. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 13 (07 1987), 501–518. doi:10.1037/0278-7393.13.3.501

[53] Bruce Schneier and John Kelsey. 1998. Cryptographic support for secure logs on untrusted machines.. In *USENIX Security Symposium*, Vol. 98. San Antonio, TX, 53–62.

[54] Richard Shay, Iulia Ion, Robert W Reeder, and Sunny Consolvo. 2014. " My religious aunt asked why i was trying to sell her viagra" experiences with account hijacking. In *CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 2657–2666.

[55] Sharon G Smith, Kathleen C Basile, Leah K Gilbert, Melissa T Merrick, Nimesh Patel, Margie Walling, and Anurag Jain. 2017. *National intimate partner and sexual violence survey (NISVS): 2010-2012 state report*. Technical Report. National Center for Injury Prevention and Control, Centers for Disease Control and Prevention, Atlanta, GA. https://www.cdc.gov/nisvs/documentation/nisvsStatereportbook.pdf

[56] Kurt Thomas, Frank Li, Chris Grier, and Vern Paxson. 2014. Consequences of Connectivity: Characterizing Account Hijacking on Twitter. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. Association for Computing Machinery, New York, NY, USA, 489–500. doi:10.1145/2660267.2660322

[57] Justin Turner. 2025. TPM Key Attestation. https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/component-updates/tpm-key-attestation Accessed: 2025-11-13.

[58] Courtland VanDam, Jiliang Tang, and Pang-Ning Tan. 2017. Understanding Compromised Accounts on Twitter. In *Proceedings of the International Conference on Web Intelligence*. Association for Computing Machinery, New York, NY, USA, 737–744. doi:10.1145/3106426.3106543

[59] Daphna Weinshall. 2006. Cognitive authentication schemes safe against spyware. In *2006 IEEE Symposium on Security and Privacy (S&P'06)*. IEEE, IEEE, Oakland, CA, 6–pp.

[60] World Health Organization. 2021. Violence against women. https://www.who.int/news-room/fact-sheets/detail/violence-against-women. Fact sheet; accessed: 2023-07-22.

[61] Ka-Ping Yee and Kragen Sitaker. 2006. Passpet: convenient password management and phishing protection. In *Proceedings of the second symposium on Usable privacy and security*. ACM, Association for Computing Machinery, New York, NY, USA, 32–43.

[62] Andrew P. Yonelinas, Michelle M. Ramey, and Cameron Riddell. 2024. Recognition memory: The role of recollection and familiarity. In *The Oxford Handbook of Human Memory*, Michael J. Kahana and Anthony D. Wagner (Eds.). Oxford University Press, Oxford, UK, 923–958. doi:10.1093/oxfordhb/9780190917982.013.32

[63] Eva Zangerle and Günther Specht. 2014. " Sorry, I was hacked" a classification of compromised twitter accounts. In *Proceedings of the 29th annual acm symposium on applied computing*. Association for Computing Machinery, New York, NY, USA, 587–593.

A AI-Generated Images and Their Distributions

We generate URID images using the SDXL-Turbo model, which produces highly diverse outputs conditioned on our structured prompts. To quantify this diversity, we computed pairwise Euclidean distances between CLIP embeddings of 100,000 generated images. The resulting distribution, shown in Fig. 5, indicates substantial variation across images and closely resembles the distribution of natural images in COCO.

For illustration, we also highlight two generated images with a CLIP distance of 1.96 — the smallest value observed. Although the embedding space considers them similar, they remain visually distinct to humans. This supports the suitability of AI-generated images as user-recognizable, low-collision identifiers.

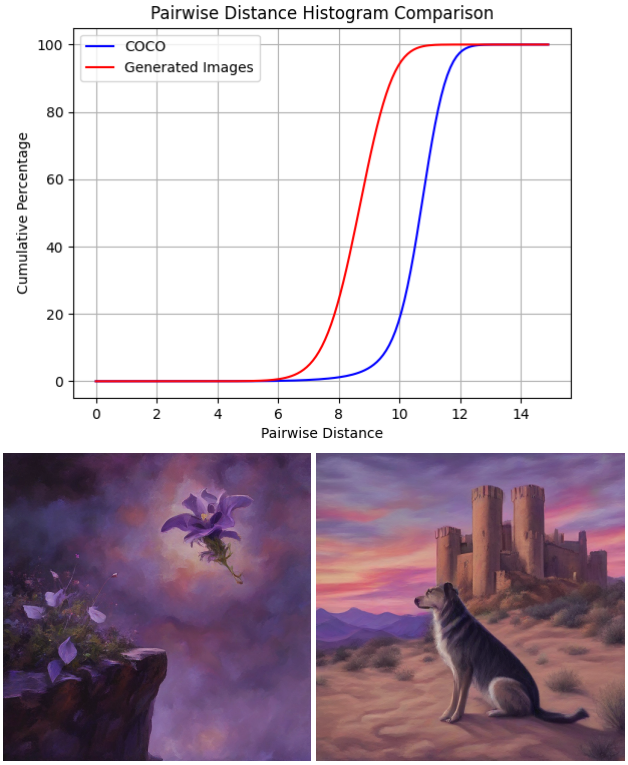


Figure 5: (Top) Percent of generated image pairs with Euclidean distance less than or equal to x, compared to the distribution with COCO image pairs. (Bottom) The closest pair of generated images (CLIP distance = 1.96).

B Single-Identifier Follow-Up Study

The main study asked participants to learn 36 identifiers, which may be more than users would realistically need to remember; in deployment, a user tracks one identifier per device. To examine recognition under this lighter memory load, we ran a follow-up study with twelve participants who had completed the main study. Each participant was assigned a single URID image and completed six brief daily sessions, each unlocking 24 hours after the previous one was completed. In each session, the participant visited one of six simulated social platforms — a different platform each day — and completed a short interaction there, such as viewing the page and leaving a comment. The assigned image appeared ambiently during these sessions, on the loading screen and embedded in the platform interface; it was never mentioned, and no task referred to it.

The recognition test unlocked immediately after the sixth session. Participants were shown ten images one at a time — their own image, placed at a random position, and nine images from the same generator that had not been shown in this study — and marked each as seen or unseen, with no indication of how many of the ten had appeared in the study. Nine of the twelve participants marked their own image as seen; 34 of the 108 other images (31.5%) were also marked seen. The own-image recognition rate (75%) is no higher than the delayed recognition scores of the main study (Section 6.3), giving no indication that the main study's 36-identifier load was the limiting factor for recognition performance.

C Security Definitions and Proofs

This appendix makes the security claims from Section 4.3 explicit and gives the definitions needed to interpret the informal arguments in the main body. Let λ be the security parameter and $\mathcal{A}$ a probabilistic polynomial-time adversary. The trusted components are the device keystore, the device attestation key, and the Privacy CA, as specified in Section 3.1. We assume the device key and CA key are EUF-CMA secure, that SHA-256 is second-preimage resistant, and that distinct identity keys are generated independently.

For an identity e, let pk_e denote the device identity public key and let

$$\mu_e = \mathrm{SHA256}(pk_e)$$

be its MRID. A protocol response is a tuple $\rho = (pk, \pi_e^{\mathrm{CA}}, ts, \sigma)$ and is accepted by the server only if $\mathsf{SP.Verify}(e, \mathrm{app}_{\mathrm{id}}, r, \rho)$ succeeds. The server maintains an outstanding set Q of tuples $(\mathrm{app}_{\mathrm{id}}, e, r, ts)$ that have been issued and not yet consumed. An authorized signing query is one that the trusted OS and urid_agent approve under the policy of Section 3.1; such a signature binds the message

$$m = \mathsf{encode}(\texttt{URID-GetSignature-v1}, \mathrm{app}_{\mathrm{id}}, e, r, ts).$$

The following definitions are the ones used in the paper's security statements.

Target-MRID Unforgeability. In the experiment $\mathrm{Exp}^{\mathrm{forge}}_{\mathcal{A}}(\lambda)$, the challenger initializes an honest device and identity e^*, and gives $\mathcal{A}$ the public transcript and the target MRID $\mu^* = \mathrm{SHA256}(pk_{e^*})$. The adversary may initialize and control other identities, obtain their certificates, observe honest responses, and request authorized signatures from the honest device. The server then creates a fresh outstanding tuple $(\mathrm{app}^*_{\mathrm{id}}, e^*, r^*)$. The adversary wins if it outputs a response ρ^* such that $\mathsf{SP.Verify}(e^*, \mathrm{app}^*_{\mathrm{id}}, r^*, \rho^*)$ accepts and the accepted MRID equals μ^*, but the honest keystore never produced a valid signature on the corresponding message

$$m^* = \mathsf{encode}(\texttt{URID-GetSignature-v1}, \mathrm{app}^*_{\mathrm{id}}, e^*, r^*, ts^*).$$

Its advantage is

$$\mathrm{Adv}^{\mathrm{forge}}_{\mathcal{A}}(\lambda) = \Pr[\mathrm{Exp}^{\mathrm{forge}}_{\mathcal{A}}(\lambda) = 1].$$

Freshness. In the experiment $\mathrm{Exp}^{\mathrm{fresh}}_{\mathcal{A}}(\lambda)$, the adversary controls the network and may observe any number of previously accepted responses. It wins if it causes the server to accept a response for a tuple that is not the current outstanding tuple or that is accepted after its timestamp window has expired. Formally, the adversary succeeds when acceptance occurs for a different $(\mathrm{app}_{\mathrm{id}}, e, r)$ after the original tuple has been removed from Q, or when the accepted response carries a timestamp outside the accepted freshness window Δ. Since the signed message includes $(\mathrm{app}_{\mathrm{id}}, e, r, ts)$ and the server enforces a single-use outstanding tuple, any such replay corresponds either to a valid forgery or a violation of the server's state machine.

MRID Uniqueness. In $\mathsf{Exp}^{\mathrm{uniq}}_{\mathcal{A}}(\lambda)$, the challenger generates two independent identity keys (pk_0, pk_1) and the adversary wins if the resulting MRIDs are equal:

$$\mathsf{SHA256}(pk_0) = \mathsf{SHA256}(pk_1).$$

The target property is that this event occurs only with negligible probability.

Cross-Identity Unlinkability. In the experiment $\mathsf{Exp}^{\mathrm{link}}_{\mathcal{A}}(\lambda)$, the adversary chooses two distinct identities $e_0 \neq e_1$. The challenger samples a bit $b \leftarrow \{0,1\}$, and then either (1) initializes both identities on one device when $b = 0$, or (2) initializes them on two different devices when $b = 1$. The challenger runs the protocol for fresh sessions and hands the resulting server-visible transcripts to $\mathcal{A}$, which outputs a guess b'. The advantage is

$$\mathsf{Adv}^{\mathrm{link}}_{\mathcal{A}}(\lambda) = \left|\Pr[b' = b] - \tfrac{1}{2}\right|.$$

The goal is that the SP cannot distinguish whether the two identities belong to one device or two, except with negligible advantage.

Proof of Theorem 4.1 (Authenticity and Freshness). Suppose the adversary causes the server to accept a response ρ^* for target MRID μ^*. Let

$$m^* = \mathsf{encode}(\texttt{URID-GetSignature-v1}, \mathrm{app}^*_{\mathrm{id}}, e^*, r^*, ts^*).$$

Because the protocol accepts only if $\mathsf{SP.Verify}(e^*, \mathrm{app}^*_{\mathrm{id}}, r^*, \rho^*) = \mu^*$ and the corresponding tuple is outstanding, the response must either (i) contain a valid signature under the honest device key on m^*, (ii) contain a certificate chain for a different public key that the server accepts as valid, or (iii) use a different certified key $pk^* \neq pk_{e^*}$ with $\mathsf{SHA256}(pk^*) = \mu^*$. Cases (i) and (ii) are ruled out by EUF-CMA security of the device key and the CA signing key, respectively: if the honest keystore never signed m^* then a valid signature on m^* under pk_{e^*} is a new forgery; if the attacker presents a CA-certified key not issued to the honest identity, then it must have forged a CA certificate or caused the CA to sign a new message outside the authorized policy. Case (iii) is a second-preimage attack on SHA-256. Therefore, if the adversary wins the target-MRID unforgeability experiment with non-negligible probability, it breaks the EUF-CMA assumption of the honest device signature or the second-preimage resistance of SHA-256.

For freshness, suppose the adversary causes the server to accept a response for a stale or mismatched tuple. The server accepts only if the signed message contains the current challenge data $(\mathrm{app}_{\mathrm{id}}, e, r, ts)$ and the tuple is still in the outstanding set Q with the current timestamp window Δ. If the accepted tuple differs from the outstanding challenge, then either the message was altered after signing, which requires a forgery on the signed payload, or the server's state check was bypassed. If the timestamp is stale, then the server rejects unless the signature is on an expired challenge, which again would require forging the signature on a different message or bypassing the freshness check. Since the message is injectively encoded with $(\mathrm{app}_{\mathrm{id}}, e, r, ts)$, a replay of an old response cannot be accepted for a different challenge. Hence any non-negligible advantage in $\mathsf{Exp}^{\mathrm{fresh}}_{\mathcal{A}}(\lambda)$ implies either a signature forgery or a violation of the trusted SP state machine. This proves the authenticity and freshness claims in Theorem 4.1.

Proof of Theorem 4.2 (MRID Uniqueness). Let pk_0, pk_1 be two independent key pairs generated by the device key-generation algorithm, and suppose $\mathsf{SHA256}(pk_0) = \mathsf{SHA256}(pk_1)$. There are two possibilities. If $pk_0 = pk_1$, then the event is a key-generation collision; because the key-generation algorithm samples from a space of size at least 2^{λ}, the probability is at most $2^{-\lambda}$. If $pk_0 \neq pk_1$, then the two distinct public keys form a SHA-256 collision. By collision resistance of SHA-256, this occurs with negligible probability. By the union bound,

$$\begin{aligned}\Pr[\mathsf{SHA256}(pk_0) = \mathsf{SHA256}(pk_1)] \\ \leq \Pr[pk_0 = pk_1] + \Pr[\mathrm{Coll}_{\mathrm{SHA256}}(pk_0, pk_1)],\end{aligned}$$

which is negligible. Therefore the probability that two independently generated public keys produce the same MRID is negligible, proving Theorem 4.2.

Proof of Theorem 4.3 (Cross-Identity Unlinkability). We define two worlds. In world $b = 0$, the challenger initializes both identities e_0 and e_1 on the same device. In world $b = 1$, they are initialized on two different devices. Let V_b denote the SP-visible transcript distribution in world b. In both worlds, each identity key is generated independently from the same key-generation algorithm, the Privacy CA certificate is generated from the same certificate-generation procedure, and the signed values are computed over fresh challenge messages with the same message format. The server's view therefore consists only of certified public keys, certificate material, signatures, and the challenge metadata; there is no device-specific value in the certificate or the signature beyond the key material itself, which is already included in the public key. Thus the induced distributions V_0 and V_1 are computationally indistinguishable, because any distinguisher would have to detect device provenance from values that are generated from the same distribution and are independent of the underlying device assignment. Formally, if an adversary could distinguish V_0 from V_1 with non-negligible advantage, then it would either break the signature scheme, the certificate distribution assumptions, or the assumed independence of the key-generation process. Since all of these are excluded by the security assumptions, the distinguishing advantage is negligible.

This proof intentionally excludes side channels (timing, IP address, account metadata, CA–SP collusion, and the degenerate case $e_0 = e_1$), which are outside the stated threat model and therefore are not part of the cryptographic guarantee. The theorem therefore captures precisely the claim made in the main text: for distinct identities, the server cannot tell whether their transcripts came from one device or two, except with negligible probability.

D Study Site Design

We designed a website to conduct the study with our participants. Some of the pages of the website are shown in Fig. 6 and Fig. 7. More details of the study are provided in Section 5.1.

E App for Generating MRIDs

We created a fully functional app that instantiates the urid_agent on an Android phone. This demonstrates that the urid_agent can be deployed without modifying the OS. This enables faster deployment of URIDs. Some screenshots of the app are given in Fig. 8.

Figure 6: From top: (1) Home page showing the current task, (2) Loading screen with URIDs, (3) Post-distraction image test, (4) Distraction task, and (5) Image-only recognition task (in the third week).

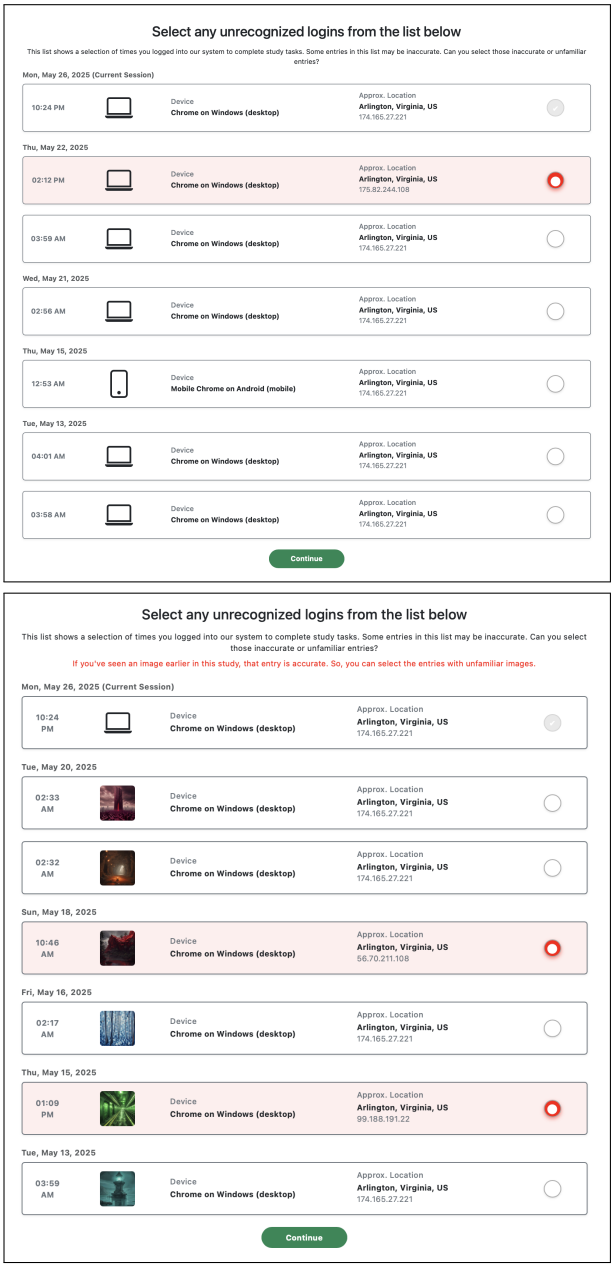
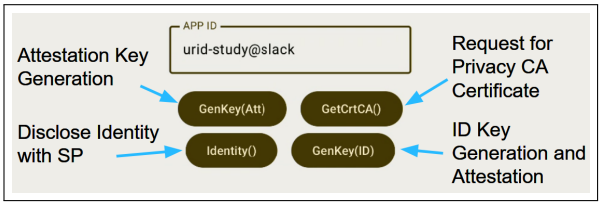
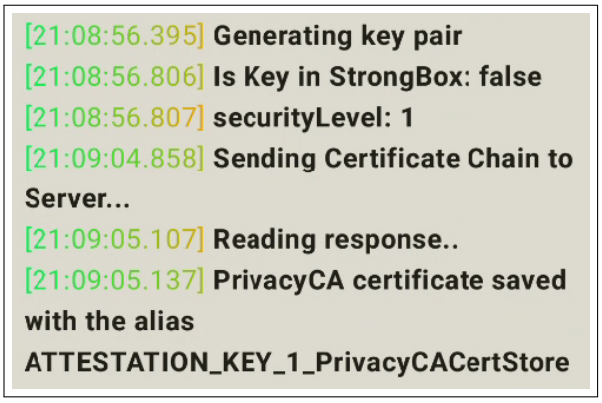


Figure 7: (Top) Control test with no URIDs, and (Bottom) experimental test with embedded URID images.





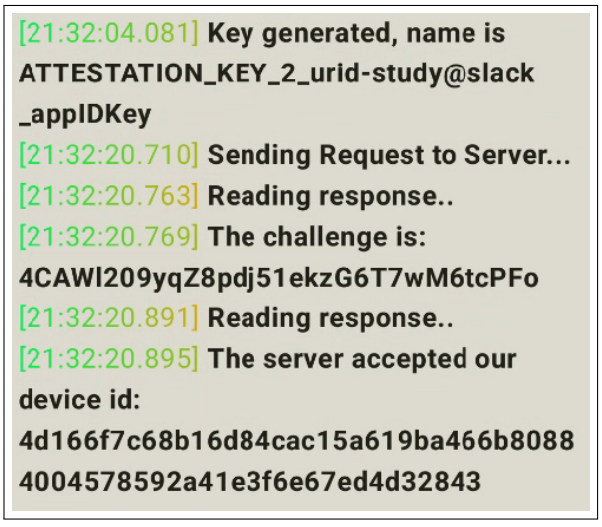


Figure 8: (Top) Android application debug interface, (Middle) Output for attestation key generation and the certificate request sent to the Privacy CA. (Bottom) Output for generating and attesting the ID key, and the ID request sent from the SP's server.