Experimental Analyses of the Physical Surveillance Risks in Client-Side Content Scanning

Ashish Hooda, Andrey Labunets[‡], Tadayoshi Kohno[†], Earlence Fernandes[‡]
University of Wisconsin-Madison
University of Washington[†]
University of California San Diego[‡]
ahooda@wisc.edu, yoshi@cs.washington.edu[†], {alabunets, efernandes}@ucsd.edu[‡]

***Abstract*— Content scanning systems employ perceptual hashing algorithms to scan user content for illicit material, such as child pornography or terrorist recruitment flyers. Perceptual hashing algorithms help determine whether two images are visually similar while preserving the privacy of the input images. Several efforts from industry and academia propose scanning on client devices such as smartphones due to the impending rollout of end-to-end encryption that will make server-side scanning difficult. These proposals have met with strong criticism because of the potential for the technology to be misused for censorship. However, the risks of this technology in the context of surveillance are not well understood. Our work informs this conversation by experimentally characterizing the potential for one type of misuse — attackers manipulating the content scanning system to perform physical surveillance on target locations. Our contributions are threefold: (1) we offer a definition of physical surveillance in the context of client-side image scanning systems; (2) we experimentally characterize this risk and create a surveillance algorithm that achieves physical surveillance rates more than 30% by poisoning 0.2% of the perceptual hash database; (3) we experimentally study the trade-off between the robustness of client-side image scanning systems and surveillance, showing that more robust detection of illicit material leads to an increased potential for physical surveillance in most settings.**

I. Introduction

Many file-sharing and communication service providers scan user image data against lists of known illicit images. This helps detect child sexual abuse material (CSAM), non-consensual pornography, and terrorist recruitment material [12], [22]. The service provider will ban associated accounts and report the user identities to law enforcement for further legal action, which depends on the severity of the violation. Various government and non-profit bodies curate the lists of illicit content, such as the National Center for Missing and Exploited Children (NCMEC) in the US, and Internet Watch Foundation in the UK.

With the rollout of End-to-End encryption of user data on services such as iCloud [2], this type of server-side scanning is no longer possible. Motivated by this, recent proposals instead perform scanning locally on the user's device (e.g., smartphone) before content is encrypted. Termed Client-Side Image Scanning (CSIS), these proposals employ perceptual hashing algorithms (e.g., PhotoDNA [31], PDQ [12]) that convert visually similar images to similar hashes. The client-side content scanning systems match perceptual hashes of user images against a curated database of illicit image hashes.

Perceptual hashes are not cryptographic hashes; they preserve the visual similarity of differing images — hashing visually similar images will produce hash values that are close to each other according to a distance metric (e.g., euclidean distance). Therefore, a perceptual hash does not change (or changes only by a little amount) when the underlying images undergo transformations like re-coding or re-sizing.

Despite the potential for this technology to curb the distribution of illicit content, there is potential for misuse. Critics of client-side image scanning have pointed out that nation states or otherwise malicious law enforcement agencies can repurpose CSIS systems to perform surveillance and censorship on many types of content [1]. For example, a nation-state could attempt to monitor the private content of whistleblowers and journalists. Consequently, a growing line of work in the community is exploring how CSIS systems employing perceptual hashing could be misused [25], [38], [1], [44].

The CSIS system involves interaction between multiple entities and uses emerging techniques such as perceptual hashing functions. It is being designed to protect against abuse from powerful adversaries. Thus, it is important to investigate various types of emerging threats from these powerful attackers. In this work, we bridge the knowledge gap about the vulnerability of CSIS to physical surveillance attacks on high-profile target individuals. As this is an emerging technology, we focus our analysis on its fundamental components and abstract away implementation-specific details when appropriate.

Our contribution is the characterization of the physical surveillance risks at the intersection of the content curators and service providers that deploy CSIS at a large scale. Specifically, we introduce a new threat model of physical surveillance in CSIS and then experimentally analyze the feasibility of the physical surveillance threat. We define physical surveillance as the capability of an attacker to visually monitor a scene. We empirically characterize the extent to which the attacker's capability approximates this idea. Our primary insight is that the photographs that users take of a scene can become accessible to an attacker who manipulates the content scanning system database.

In our threat model, the attacker is a government or nation

Network and Distributed System Security (NDSS) Symposium 2024
26 February - 1 March 2024, San Diego, CA, USA
ISBN 1-891562-93-2
https://dx.doi.org/10.14722/ndss.2024.241401
www.ndss-symposium.org

state interested in conducting physical surveillance on a high-value target such as a suspected whistleblower, journalist or activist. These people might take photographs of themselves or of their surroundings. Due to the presence of CSIS on their phones, each image will be matched against a database of illicit image hashes. If the attacker can strategically manipulate this database of illicit hashes, then they can induce matches that would result in the target's images being transmitted to the service provider and getting decrypted. With the raw images, the attacker gains a new surveillance capability with lower requirements for on-site work compared to placing a camera at that location. The caveat is that this surveillance capability only approximates a real camera because the images they get access to depend on the photographs that users take and the inexact matching process. Our work experimentally characterizes the extent to which such a physical surveillance attack is possible.

At this point, the attacker can perform additional analyses on the images to extract different types of information. For example, they could run the image through a face recognition system [33], [6], [52]; or if the CSIS system also reports user identity, they can automatically place a specific user at a specific physical location.

Compared to traditional camera-based surveillance, physical surveillance in CSIS is an alternative monitoring approach: CSIS-based scanning relies on different assumptions about the adversary's capabilities and goals. Note that installation of a hidden camera at the target location might require specialized equipment, which could compromise an employee carrying it. Additionally, after the camera is found, it is possible to stop and attribute an attack. On the other hand, the CSIS-based monitoring capability can provide a stealthy and uninterrupted monitoring approach for specific remote or high-risk locations. We show that this form of monitoring can even be possible with no physical access to the target scene at all, based only on publicly available images and videos of the location, such as 3D tours provided by hotels, real estate, or homeowners instead [30].

A possible way to carry out CSIS-based surveillance is to backdoor the CSIS system's perceptual hashing model directly [24]. However, if a machine learning model is deployed on the client-side, this publicly exposed model will be under constant scrutiny and can be reverse-engineered by security researchers [49], [45], [46]. Additionally, if a service provider wants to replace a CSIS system with an updated, more performant version, this can destroy the backdoor functionality, therefore an adversary will need to maintain control over the release process as well. In contrast with hash function backdoors, our work exploits the private, persistent, and opaque nature of the CSAM database: CSAM database is considered highly confidential [1], and the content inside it is unlikely to be removed. Given that this database is a set of non-invertible binary hashes of illicit content, governments could even try to compel a service provider to alter the system functionality [23] by inserting opaque hashes, followed by a wiretap order to read all subsequent matching images.

The attacker's high-level strategy is to poison this curated hash database of illicit content with hashes crafted to match the images that users might take at target physical locations. This requires overcoming a few challenges. First, perceptual hashing-based CSIS is robust to only small input image modifications. Physical surveillance requires detection of a wide variety of images of a scene. Second, the attacker's poison hashes have to correspond to illicit content. Otherwise, a human curator of the database can easily flag images submitted by the attacker as being irrelevant.

We address these challenges in the following ways. First, we contribute an algorithm that computes the optimal set of poison hashes using a Gaussian Mixture Model (GMM) to approximate the image hash distribution at the target physical scene. Our poisoning attack is agnostic to the perceptual hash algorithm and is effective under a wide range of environmental conditions. Second, we rely on the observation that perceptual hashing algorithms are susceptible to adversarial examples [44], [25], [38], [10]. It is possible to craft two images that are visually different but their hashes are very similar. Specifically, starting from a known illicit image, we modify it so that it still looks like illicit content to a human curator but its corresponding hash will collide with the hashes of a specific physical scene.

A final challenge is the experimental setup and datasets required for such an analysis. We do not experiment with illicit images, but rather, use existing datasets from the ML community to approximate the concept of illicit content. Specifically, we use two disjoint subsets from the Imagenet dataset to categorize illicit and benign images [50]. We also collect images of multiple target physical locations via scraping publicly available user images uploaded on Instagram and also manually capturing photographs [17].

We focus our analyses on two classes of perceptual hash functions: Non-Learning based and Learning based. For non-learning based, we use PDQ [12] that is currently deployed as a server-side scanning system, but it is useful in a client-side deployment scenario as well. For learning based, we use a contrastive learning-based feature extractor from Self-Supervised Copy Detection (SSCD) [37]. We use random projection to convert the feature vectors to 256-bit hashes. We will refer to this perceptual hash function as SSCD-Hash. Apple's proposal of a commercial client-side scanning system also used a learning-based NeuralHash function [22].

Contributions. Our work identifies a new type of surveillance threat and provides experimental evidence demonstrating the feasibility of physical surveillance, should CSIS systems be deployed to the public at a large scale.

1) We identify a gap in the current risk assessment of CSIS and provide a novel definition of physical surveillance threat model for perceptual hashing-based client-side image scanning systems. We introduce a poisoning attack that can covertly re-purpose CSIS systems to perform physical surveillance. We propose a Gaussian Mixture Model (GMM) based approach to find the set of poison hashes.
2) We characterize the attack on real user data collected from Instagram. We also evaluate the feasibility of our attack under a diverse set of environmental conditions. Overall, our attacks are able to achieve $> 30\%$ surveillance by poisoning just 0.2% of the illicit content database, without significantly changing the false positive rate of the system.
3) We characterize the trade-off between the system's ability to detect the designated illicit content and the potential to

perform covert physical surveillance: more robust detection of illicit content leads to higher risks of physical surveillance.

Ethical Considerations. Our work informs the conversation around the benefits and risks of client-side image scanning technologies. Specifically, we contribute a set of experimental analyses that showcase the extent to which an adversary might misuse this technology to perform physical surveillance. We do not take a concrete stance on the issue of whether CSIS technology in its current form should be widely deployed; nor do we wish to imply that because physical surveillance is possible, such technology should never be deployed. Rather, we agree that curbing illicit content like CSAM does require technological innovation, but it has to be balanced with an understanding of the inherent risks. We will not be releasing attack code publicly but will manage requests on a case-by-case basis for sharing the code in the interest of scientific exploration. For our evaluation, we scraped images uploaded to Instagram by public user accounts. We did not store any raw images and performed experiments only on the image hashes. We provide more details on the data collection procedure in Section V-B. Finally, we will not be releasing the Instagram dataset publicly.

II. Background

A. Client-Side Image Scanning Overview

Client-side image scanning (CSIS) systems are part of the public debate around whether law enforcement should have special access to plaintext communications. A recent shift towards end-to-end encrypted communications has created barriers for law enforcement and service providers to detect illicit content, such as child exploitation and terrorism imagery. Client-side Image scanning is designed to selectively relax end-to-end encryption guarantees depending on the content being transmitted in an encrypted channel. By scanning content on the client device before it is encrypted, CSIS allows service providers and law enforcement to open up encryptions if the underlying content matches known illicit content. At this point, the content provider can report the user's identity to law enforcement for further investigation.

B. Client-Side Image Scanning and Perceptual Hashing Formal Definition

A naïve approach to finding examples of illicit images on a device would be to search for exact matches of those examples using a hash function, such as a cryptographic hash function $\mathcal{H} : \mathcal{X} \to \{0,1\}^n$, where $\mathcal{X}$ is raw image data and n is a hash size. However, such a system would only find exact matches of the content and would be trivially bypassed due to collision resistance of the cryptographic hash functions: small changes in the input $\mathcal{X}$ will result in a different value of the output hash. Instead, current systems use a different type of hashing, perceptual hashing, with some degree of invariance to match examples that look visually similar to humans. We formally define the CSIS and its properties below, similar to the existing works on perceptual hashing [38], [44], [25].

For the raw image $\mathcal{X}$ and a length n, a perceptual hashing function is defined as $\mathcal{P} : \mathcal{X} \to \{0,1\}^n$. Here the length of the bit string depends on the type of perceptual hashing algorithm.

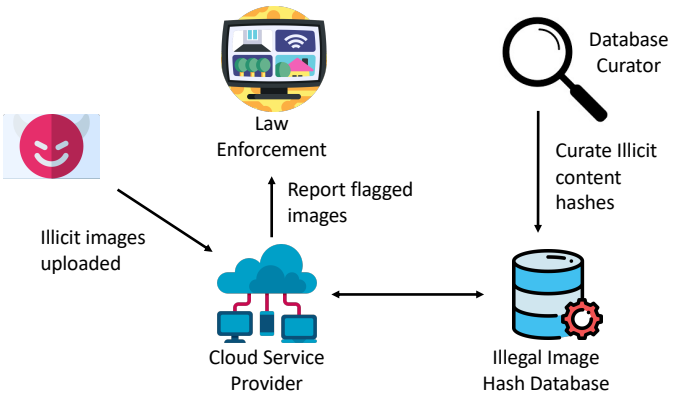


Fig. 1. Pipeline of a CSIS system. The Content Curator collects illicit images from multiple sources such as hosting providers, and maintains a database of illicit content hashes. This database is sent to the service provider to enable CSIS. Illicit images uploaded to the service provider are detected by matching against the illicit hash database and may go through an additional review process. Finally, the flagged cases are reported to Law Enforcement agencies.

PDQ produces 256-bit hashes. For consistency, we design the SSCD based perceptual hash function to also produce 256 bit hashes. Similarity between perceptual hashes is represented by a distance metric, $\mathcal{D} : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$, which is typically hamming distance. A CSIS system $\mathcal{S}$ is defined by a Perceptual Hashing function $\mathcal{P}$, a distance metric $\mathcal{D}$, a threshold $t \in \mathbb{R}$, and a database $\mathcal{C} \subset \{0,1\}^n$ of hashes for the illicit content examples.

CSIS systems detect illicit images by matching perceptual hashes of a user's images against a database of hashes $\mathcal{C}$ computed using a curated list of illicit images (CSAM etc). When a user uploads an image $\mathcal{X}$, the CSIS system flags it if the image's perceptual hash has a distance less than a threshold t from at least one hash $c \in \mathcal{C}$ in the database, i.e., $\mathcal{D}(\mathcal{P}(\mathcal{X}), c) \leq t$. Microsoft's PhotoDNA [31], and Facebook's PDQ [12] are notable examples of perceptual hashing algorithms used for image scanning.

Even though Client-Side Image Scanning Systems (CSIS) are only designed to flag illicit content, the operational landscape is considerably more complex it is less clear what level of access to images or hashes each party is granted. The standard operation of CSIS involves a triad of entities: the Content Curator, who maintains a database of illicit content; the Service Provider, who deploys the protocol within a messaging application; and Law Enforcement (Figure 1). Content curators are usually non-profit organizations such as the National Center for Missing and Exploited Children (NCMEC) in the US and the Internet Watch Foundation in the UK, and they obtain illicit material, calculate perceptual hashes, and relay them to the Service Provider. Next, the Service Provider's CSIS system attempts to match user-uploaded photos against the database and may subject flagged photos to an additional review process to deal with false positives, a common occurrence in any perceptual hashing task. If validated, these flagged photos may then be reported to Law Enforcement. Consequently, both flagged private images and the list of illicit content hashes could potentially be accessed by parties beyond the Service Provider, creating a multi-layered, interconnected web of information.

Despite the wide-ranging functionality of CSIS systems, its concrete instantiation takes many forms, varying significantly based on the regulatory regimes, country-specific laws, and technological infrastructure in play. Given the array of operational variations, it becomes critical to maintain a broader lens when assessing the privacy and security risks associated with CSIS. Therefore, in this paper, instead of focusing on any particular CSIS proposal, we center our threat modeling and analysis on the fundamental components inherent in all CSIS configurations. This broader approach is intended to provide a more comprehensive understanding of the systemic vulnerabilities and risks associated with CSIS, thereby offering valuable insights applicable across various CSIS systems, irrespective of their specific contextual or operational nuances.

C. Threats against Content Scanning

The landscape of security studies on Client-Side Image Scanning Systems (CSIS) is growing yet still somewhat limited in its scope. Prior works have illuminated three principal threat categories linked to CSIS: extraction of source images, evasion of detection, and triggering false detection. In the first instance, attackers with access to perceptual hashes can invert them to retrieve the source images of illicit content, which due to their sensitive nature, are not publicly distributed [3]. Concurrently, adversaries can deploy either gradient-based or gradient-free techniques to subtly modify an original image, enabling the circumvention of CSIS and the distribution of illicit content [25], [38]. The third threat involves the creation of images whose hash matches the ones in the database, resulting in the wrongful flagging of private user images. These collision-based threats, heightened by the possible compromise of a service provider, can lead to the deanonymization of users, the detection of censored images, and even the framing of innocent users for trafficking illicit content. In this work, we focus on the third threat category.

Existing work on collision-based threats primarily focuses on the surveillance of digital content. Our research, however, significantly expands this understanding by introducing a new threat dimension – physical surveillance. We define physical surveillance as the potential for an adversary, especially state actors or those with significant capabilities, to exploit hash collision attacks to perform surveillance in the physical world through the CSIS system. This extends the surveillance risks beyond the digital realm and into our physical spaces. This novel conceptualization of surveillance risk represents the primary focus of our paper, thus offering a significant advancement in the understanding of surveillance possibilities through CSIS.

III. Physical-world Surveillance

We discuss a model for physical surveillance by misusing client-side image scanning technology. Concretely, the attacker wants to monitor a physical scene in a way that approximates them installing a camera at that scene. Our core insight is that users with a CSIS system deployed on their smartphones will function as an "intermittent crowd-sourced camera."

Attacker Motivation. The attacker is a government or nation-state interested in conducting physical surveillance on a high-value target such as a suspected whistleblower, journalist or activist. The goal for doing this type of surveillance is well-documented and can include trying to determine the identity of secret informants or controlling the flow of information that can harm the nation state's agenda if made public [32].

For an attacker interested in performing physical surveillance, the traditional method involves the installation of a physical camera at the targeted location. However, this approach has several drawbacks. The deployment of a physical camera requires specialized equipment and often necessitates collaboration with individuals who have access to the target site. This method can be expensive, intrusive, and may risk detection. In addition, physical cameras aren't easily scalable, especially across multiple locations. Thus, the need arises for a more covert and scalable surveillance method. Using Client-Side Image Scanning Systems (CSIS) as an "intermittent crowd-sourced camera" presents a promising alternative that overcomes these challenges.

Modifying CSIS for surveillance could take two broad approaches — backdooring the perceptual hash or poisoning the illicit content database. The backdooring approach involves tampering with the perceptual hashing function [24]. This approach presents a different set of trade-offs to the attacker. The perceptual hash is publicly accessible and any changes made to it could be detected during updates or routine checks. Additionally, the backdoored perceptual hash must remain stable despite possibly frequent updates to the algorithm — a routine occurrence because service providers are always trying to improve the performance of their algorithms. By contrast, poisoning the illicit image database is less prone to detection. The database is confidential and consists of opaque, non-invertible hashes of illicit content. An audit of the database for potential misuse or forensic analysis would be slow and challenging, making it a convenient and stealthy target for poisoning.

With the illicit content database poisoned, an attacker can effectively turn the CSIS into an 'intermittent crowd-sourced' camera. This attack requires only a limited number of photos of the target location and potentially no physical access to the site. By exploiting this vulnerability, every user unknowingly contributes to the attacker's surveillance network, thus creating a 'crowd-sourced' camera. This method is adaptable across different perceptual hashing algorithms, as it primarily targets the illicit content database. We believe that this offers a better set of trade-offs for nation state attackers who wish to perform physical surveillance, compared to existing techniques in CSIS mis-use.

Threat Model. We observe that since CSIS is an emerging technology with many possible implementations in different types of regulatory environments, it is important to abstract implementation/deployment details away so that we can examine threats on a fundamental level. Therefore, we develop our threat model according to the basic components that any CSIS should have, independent of deployment details.

We assume that the attacker has the following abilities:

- *Scene Images*: Obtain images of the target scene. Our experiments indicate that ~ 500 target scene images are sufficient for most settings (see Section V-B).
- *Illicit Images*: Obtain or generate new illicit-looking images.

- *CSIS Database Access*: Poison the CSIS Illicit Image Hash Database. Our experiments indicate that adding $\sim 0.2\%$ additional hashes is sufficient (see Section V-C).
- *Flagged Images Read Access*: Read illicit images flagged by the CSIS system.

When designing a technology, we argue that it is vital to consider an upper bound on adversarial capability, especially when the exact details of an adversary's capabilities are unknown. In doing so, if one determines that the system is secure under such assumed adversarial capabilities, then it will also be secure against adversaries with less capabilities. On the other hand, if the system is insecure under such capabilities, and if such capabilities are plausible, then there is cause for alarm. With this context in mind, next, we discuss why it is reasonable to assume that high power adversaries, including governments and nation states, might have the above capabilities.

Access to the target scene and illicit image examples: The attacker has physical or virtual access to the surveillance location and is able to obtain photographs and videos of it. In the case of physical access, the adversary can record the location ahead of time. In the case of scenes like hotel rooms, physical access may not be easily available but the attacker could obtain the necessary photographs and videos from the 3D tours, provided by the hotel, real estate web sites or home owners [30]. We assume that obtaining illicit-looking images is not hard because a nation state could have acquired such data through its law enforcement agencies. Additionally, the adversary can source images from CSAM content leaked in the past or rely on AI image generation tools [20].

CSIS database access: There are multiple methods by which an adversary could introduce poisoned images into the CSIS database. The options vary in the amount of influence the attacker can exert.

- Compromise CSIS database supply chain: The attacker tricks the content curator into inserting perturbed copies of illicit images into the database. These perturbed copies are designed such that their perceptual hashes match the poison hashes that the attacker wants to add to the hash database but visually these images look like illicit content. Concretely, this can happen in the following ways: (1) illicit content is sourced from unverified hosting providers, where an adversary injects malicious content [18] (2) content curator obtains illicit content directly from an adversary who files a report to the curator. We assume that the content curator visually validates any submitted image as being illicit content before its hash is added to the CSIS database.
- Influence the CSIS database content curator: A step up in privilege level, the government/nation state compels the content curator to modify the hash database by inserting specially crafted opaque hashes. The attacker achieves this using legislative, financial, or diplomatic methods. In this case, the attacker can insert a much larger number of poison hashes.

Flagged images read access: Attacker can access the images that the CSIS system matches with the database of illicit content. Given the diverse implementations of CSIS across different countries and legal jurisdictions, this could happen through different mechanisms:

- Nation state creates legislation that requires the service provider to report all matched images. This legislation could be a requirement that the service provider has to accept in order to conduct business in the nation state's jurisdiction. It could also be implemented via legal tools such as wiretap orders or subpoenas, depending on the regulatory environment. No matter the method of enforcing the requirement, the outcome is that the matched images are decrypted automatically — a feature of the CSIS protocol. These decrypted images are then forwarded to the attacker without any processing, thus ensuring the capability of *flagged images read access*.
- A covert government agent works as an employee inside the service provider to consult on the CSIS system. It is possible that the government bribes such an employee and convinces them to cooperate: company positions with access to sensitive user data are advantageous for nation states who want recruit an insider agent to spy on dissidents [34]. In this case, the inside agent could be instructed to forward all matching images to the attacker.

In both cases, a service provider might monitor global statistics, such as the number of images being decrypted due to matches with the CSIS database. If there a spike in this number, it could alert the service provider about the possibility of an attack. However, we observe that any image matching system has a natural false positive rate. The attack can hide inside this natural rate. As we will show in Section V-D, our proposed attack does not significantly change the natural false positive rate of the system.

IV. Design

Our poisoning attack for physical surveillance consists of two components corresponding to the challenges discussed above: (1) computing an optimal set of poison hashes that addresses the issue of weak perceptual hash function robustness to image transformations; (2) computing a set of images that appear like illicit content to a human curator but whose perceptual hash values are close to hashes computed in step 1 (i.e., a set of poison delivery images). We observe that step 1 of our attack framework for computing poison hashes does not depend on perceptual hash function details. The second step of computing the poison delivery images depends on whether the perceptual hash function is differentiable because the algorithm computes adversarial examples. If the hash function is not differentiable, we utilize gradient-free methods to compute the adversarial example. Figure 2 describes the interacting entities and the multiple components of the attack pipeline.

A. Crafting Poison Hashes

Poisoning attacks against ML models insert malicious images to the training data which corrupts the training process and modifies the model weights [43], [40]. However, in the CSIS setting, the perceptual hash function does not depend on the database of illicit images (PDQ is not a deep learning model and SSCD-Hash is trained on publicly available images and not the illicit images or hashes). In contrast to other poisoning attacks, our attack exploits the transformation invariance of perceptual hashing. Perceptual hash algorithms are invariant to small image transformations such as scaling, cropping, and recoloring. They are designed to detect whether

two images are syntactically similar. Note that this is different than semantic similarity which depends on high-level features of the image content. This is also evident from the fact that these hash functions are not robust to semantic transformations like rotation and translation. For instance, rotating an image by only 5 degrees changes more than 10% of the hash bits for both PDQ and SSCD-Hash [9]. This fairly limits the ability of a CSIS system to detect transformed versions of illicit images. This is more so the case for physical surveillance which encounters high degrees of semantic transformations such as changes in perspective and viewing angles. To address this challenge, we exploit a key property of a CSIS system – Detection of an image only depends on collisions with the best matching hash in the illicit image database. This means that regardless of the perceptual hash algorithm, a CSIS system can potentially be robust to semantic transformations as long as there is at least one hash in the illicit image database corresponding to every transformation instance. Consequently, a CSIS that uses PDQ or SSCD-Hash can be designed to robustly detect rotations of a specific image if we add the hashes of all possible rotations of the image to the illicit image database. However, this would not scale well to semantic transformations involved in arbitrary photographs of a scene which constitute of combinations of perspective, viewing angles, and even environmental conditions. In this case, an adversary would need to poison the illicit image database with an unbounded number of hashes to account for the continuous physical transformation space. This is not feasible due to two reasons - (1) Increasing the illicit database size of a CSIS system increases the false positive rate [25], and (2) Each hash needs to be inserted without raising suspicion from the database curators. Therefore, adding a very large number of hashes to the illicit database would make the system unusable (which will be detected by the curators) and also incur a lot of effort from the adversary. A practical physical surveillance attack needs to work with a limited number of poison insertions. Note: Inserting poisons corresponding to semantic transformations of a scene allows the CSIS system to robustly detect images from only that specific scene, and the CSIS system is only syntactically robust to all images outside that scene.

Finding the optimal set of hashes. As described in Section II, every CSIS system has the following four components: Perceptual Hash Function $\mathcal{P}$, illicit database $\mathcal{C}$, distance metric $\mathcal{D}$, and a distance threshold t. Any image $\mathcal{X}$ is flagged as illicit if there is at least one hash $c \in \mathcal{C}$ that is closer than t from the image hash, i.e. $\mathcal{D}(\mathcal{P}(\mathcal{X}), c) < t$. Therefore, detection works as long as the distance is less than t. We use this property to reduce the number of poisons required to perform physical surveillance. In order to formalize this, we first need a way to represent all possible photographs of a scene. For a scene S, let G_S be an image generating function where $I \xleftarrow{\$} G_S$ represent an image captured from the scene S. To successfully perform physical surveillance on scene S, the CSIS system should be able to detect all images generated by G_S. The task of generating the poisons for a CSIS using perceptual hashing: $(\mathcal{P}, \mathcal{D}, t)$ can be modeled as finding the optimal set of k hashes, $\mathcal{U}_k$ such that :

$$\arg\max_{\mathcal{U}_k} \mathop{\mathbb{E}}_{I \xleftarrow{\$} G_S} \left[\mathbb{1} \left(\min_{h \in \mathcal{U}_k} \mathcal{D}(h, \mathcal{P}(I)) < t \right) \right] \tag{1}$$

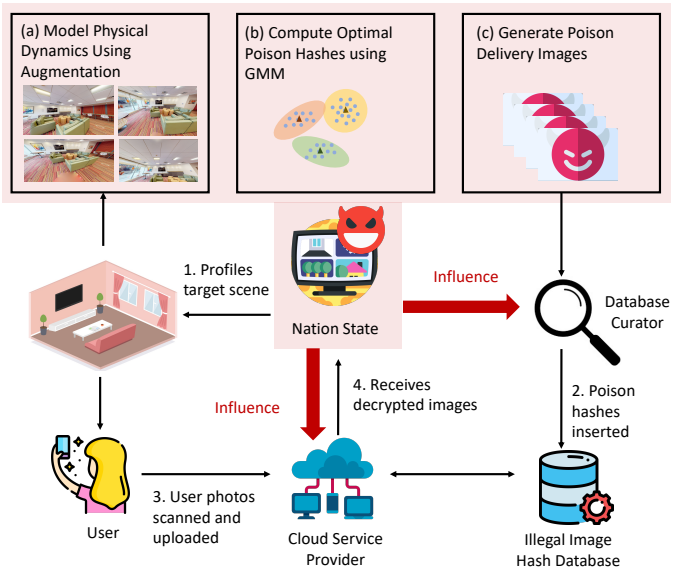


Fig. 2. Pipeline of our physical surveillance attack. The attacker profiles a target scene ahead of time, computes a set of poison hashes using the GMM approach, and finally inserts them into the illicit hash database using a set of crafted poison delivery images. Unwitting users take photos at the target scene that then collide with the poison hashes, resulting in the attacker gaining plaintext access to those images.

Here, the inner minima finds the best matching hash for a specific image I and the indicator function outputs a 1 or 0 based on whether image I is detected by the set of poisons $\mathcal{U}_k$. In contrast, the outer maxima finds the best set of poisons that in expectation, maximizes the detection of images generated from G_S. Equation 1 is a modified version of the covering code problem [7]. The covering code problem finds a set of code-words in a space with a property that every other element in that space is within a fixed distance from some code-word. This problem has applications in data compression and error correction. Equation 1 aims to find a probabilistic covering code for the distribution of images generated by G_S. The original covering code is an NP-complete problem [16], which makes Equation 1 even harder to solve. Therefore, we solve for an approximate solution of the problem using Gaussian Mixture Models (GMMs).

Gaussian Mixture Models: Gaussian Mixture Models (GMMs) represent a class of probabilistic models that assume all data points in a dataset are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. Let $x_1, x_2, ..., x_N$ be the set of input data points. A Gaussian Mixture Model is typically defined by two primary sets of parameters: the mixture weights and the component Gaussian parameters. The mixture weights $\pi_1, \pi_2, ..., \pi_K$ sum to 1 and represent the proportion of the total data represented by each Gaussian. The GMM is comprised of K Gaussian components where the k-th Gaussian is parameterized by its mean vector μ_k and covariance matrix Σ_k. The probability of a data point x_i is given by:

$$p(x_i|\pi, \mu, \Sigma) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k) \tag{2}$$

where $\mathcal{N}(x_i|\mu_k, \Sigma_k)$ is the probability density function of the multivariate Gaussian distribution with mean μ_k and covariance matrix Σ_k.

The goal of GMM-based clustering is to estimate the parameters π, μ, Σ of the GMM given the observed data. This is typically achieved using the Expectation-Maximization (EM) algorithm. In the E-step of the EM algorithm, the posterior probabilities, w_{ik}, of each data point x_i belonging to the k^{th} Gaussian is calculated using Bayes' theorem:

$$w_{ik} = \frac{\pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_i|\mu_j, \Sigma_j)} \tag{3}$$

In the M-step, the parameters of the Gaussian distributions are updated to maximize the expected log-likelihood of the observed data:

$$\mu_k^{new} = \frac{\sum_{i=1}^{N} w_{ik} x_i}{\sum_{i=1}^{N} w_{ik}} \tag{4}$$

$$\Sigma_k^{new} = \frac{\sum_{i=1}^{N} w_{ik}(x_i - \mu_k^{new})(x_i - \mu_k^{new})^T}{\sum_{i=1}^{N} w_{ik}} \tag{5}$$

$$\pi_k^{new} = \frac{\sum_{i=1}^{N} w_{ik}}{N} \tag{6}$$

The E-step and M-step are iteratively executed until the log-likelihood of the observed data under the estimated GMM converges. We use a GMM with K components which is equal to the number of poison hashes. The set of mean values of all the components in the GMM is used as the poison hashes.

First, we hash each of the captured images to get a set of scene hashes $\{\mathcal{P}(s_1), \mathcal{P}(s_2), ..., \mathcal{P}(s_n)\}$. Each hash in this set can be viewed as a l dimension data point of binary categories, where l is the length of the hash bit string (which is 256 for both PDQ and SSCD-Hash). We now initialize the GMM with k-means and then iteratively update the component parameters. Since the final poison hashes need to be bit strings (and not real vectors), we round the mean value estimates of each component to a binary vector at the end of each iteration. Once we have computed the set of optimal hashes, we next look at how to insert these hashes into the illicit database.

B. Computing Poison Delivery Images

Once the optimal set of hashes is computed, they need to be inserted into the illicit content database. Each of the optimal hashes returned by the GMM algorithm is likely to correspond to a scene image. We can submit the scene images directly to the curator for insertion into the illicit database, but this is unlikely to succeed as the scene images do not fall into any of the illicit categories and therefore, are likely to be rejected by the curator during a manual review. We get around this constraint by submitting images that perceptually look like illicit content but get hashed to a poison value — we term these as *poison delivery images*. This can be achieved by adding adversarial perturbations to a known illicit image such that it now hashes to a poison hash computed from Step 1 above. Basically, the task is to find an image x' which is perceptually similar to a known illicit image x i.e. $||x' - x||_2 \leq \epsilon$ for some small ϵ and $\mathcal{P}(x') \approx h$ for $h \in \mathcal{U}_k$. Here, we borrow from existing work on generating adversarial perturbations to cause collision attacks on perceptual hashing functions [25], [10], [44], [38].

The CSIS system requires the used perceptual hash function to be public (since it needs to be deployed to the user device). Therefore, we can utilize gradient-based white box optimization techniques like Projected Gradient Descent to compute the delivery images. These techniques are especially effective against deep learning-based perceptual hash functions such as SSCD-Hash. Some non deep learning based perceptual hash functions such as PDQ employ non-differentiable functions such as quantization and median. They are, however, vulnerable to iterative attacks utilizing zero order gradient estimation (such as Natural Evolutionary Strategies [53]) and also attacks involving reverse-engineering the components of the hash function.

We note that the CSIS system may also derive the illicit hashes from the intersection of content curators belonging to separate sovereign jurisdictions, to reduce any control by government authorities. In this case, the perturbed plain text illicit image needs to be submitted to both curators.

V. Experimental Analyses of Surveillance Risks

We perform several experiments to demonstrate the feasibility of targeted physical surveillance. We explore surveillance success rates for various system parameters and environmental conditions. Finally, we characterize the interplay between physical surveillance and illicit image detection and find that more robust detection of illicit material leads to better surveillance success rates for the attacker.

A. Overview

Our evaluation answers the following questions:

Q1. How effectively can an adversary conduct physical surveillance for targeted locations?
We demonstrate that our poisoning attack achieves a high surveillance rate for 6 different locations relative to a false positive rate of only 0.2% (i.e., the probability that a benign image's hash matches an entry in the poisoned illicit hash database). When the adversary has physical access to capture the scene images, our attack achieves a surveillance success rate of 61% (average of 36.2% across all scenes) for SSCD-Hash and 31% (average of 23.7% across all scenes) for PDQ, where we define success rate as the fraction of user photographs that get decrypted due to collisions with the poison hashes. Further, we show that the attack is highly targeted and only increases the surveillance rate of the target location.

Q2. How does the attack affect the false positives in the underlying CSIS from the viewpoint of the service provider?
Our attack is highly targeted and has minimal effect on the false positive rate of the CSIS. Our attack achieves an average surveillance rate of 36.2% (SSCD-Hash) and 23.7% (PDQ) for the target scenes with a false positive

rate of only 0.2%. Furthermore, when considering a dedicated dataset of benign scene images, our attack minimally affects the false positive rate – 0.412% to 0.474% (SSCD-Hash) and 0.218% to 0.267% (PDQ).

Q3. How do algorithm parameters affect surveillance performance?
Our attack's optimal hash selection outperforms and provides a relative improvement of around 47% compared to a baseline random hash selection strategy on an average across all location settings. The surveillance rate of our attack increases as the adversary is able to add more poisons varying from 15% (for 500 poisons) to 53% (for 5000 poisons).

Q4. How do the environmental factors affect the surveillance performance?
The performance of our attack decreases when there are unseen variations to the scene's environmental conditions (e.g., changes in furniture or lighting). However, it is still able to obtain 32% of images with unseen variations to scene layout and 36% under unseen lighting conditions (as compared to 57% under minimal variations).

Q5. How does the surveillance rate compare with the natural performance of the CSIS system? How can the design of the CSIS system be modified to reduce surveillance risks?
CSIS systems can detect larger variations of illicit images as the perceptual hashing function threshold increases. However, as the threshold increases, the surveillance rate grows faster than the CSIS performance in most settings. Consequently, modifying the current CSIS systems to prevent surveillance would severely limit the system's ability to detect inexact illicit images.

B. Experimental Setup

A challenge in understanding the surveillance risks in CSIS is determining an appropriate setup without access to illicit content. We adopt the following settings to evaluate our poisoning attack for physical surveillance.

Client Side Scanning Parameters. As we do not have access to illicit images, we evaluate our attack using the ImageNet dataset. We use two mutually disjoint subsets of ImageNet to approximate the concept of illicit and benign images. From the illicit subset, we further select two mutually disjoint sets – (1) to construct the CSIS database $\mathcal{C}(|\mathcal{C}| = 500k)$ and (2) to represent the set of illicit images available to the attacker, $X_{inject}(|X_{inject}| = 5k)$. The size of X_{inject} cannot be less than the number of poisons since each delivery image needs to be a different illicit image. Therefore, we set the size of X_{inject} to be $5k$ (the maximum poison budget used in our evaluation). We use $10k$ benign images to evaluate the false positive rate of the CSIS system which acts as a baseline.

Scene Image Dataset. We evaluate our attack on 6 physical locations. Four of these locations are popular tourist spots – The Leaning Tower of Pisa (Italy), The Pyramids at Giza (Egypt), Stone Henge (United Kingdom), and Lennon Wall (Czechia). For each of these locations, we scrape user photographs from Instagram using an alias account. We search for images of a given location using HashTags: # <location name>. Our alias account was not a "follower" of any private account[1]. Therefore, our search queries only returned images from public accounts. Furthermore, we also perform a manual validation on the collected images to remove any image not captured at the specified location setting. For each location, we split the validated images into two disjoint sets – (1) Reference Set: scene images available to the adversary, (2) User Set: scene images uploaded by the user. The details for the number of images in each set is shown in Table I.

Phys. Location	|Ref. Set|	|User Set|	Ref. Source	User Source
Pisa Tower	~ 600	~ 150	Instagram	Instagram
Pyramids Giza	~ 700	~ 200	Instagram	Instagram
Lennon Wall	~ 200	~ 100	Instagram	Instagram
Stone Henge	~ 500	~ 150	Instagram	Instagram
Room 1	~ 45000	~ 10000	Sam. S22	Pixel 7 Pro
Room 2	~ 45000	~ 10000	iPhone 12	Sam. Z Flip 4

TABLE I. DATASET FOR PHYSICAL SURVEILLANCE POISONING ATTACK.

We selected the remaining two locations such that we could physically capture photographs of the scene. These were two indoor settings - Room 1 and Room 2. Two of the authors performed data collection from the attacker and the user perspective respectively. For each setting, the attacker captured multiple 6-minute videos scanning the entire scene. Similarly, the user also captured multiple 2-minute videos in the room. We use the frames from these videos to be the Reference and User set respectively. The two authors collected this data using two different phone cameras since the attacker would not know the camera configuration for all the users. To evaluate different environmental conditions, we performed this data collection under 3 different lighting settings. We also collected data for 3 different room layouts where we moved the objects and furniture. Finally, we collected an additional user video where another author was in the field of view to account for any unseen moving objects or people in the user images. Details on the collected data are presented in Table I. The number of images collected for the physical settings - Room 1 and Room 2 are significantly higher than the Instagram settings since in the latter, we are limited by the public images available for each specific location.

We ensured that only the authors were present in the recordings of Rooms 1 and 2, and thus, we did not need IRB approval. No other individuals were incidentally recorded in any datasets discussed above. Additionally, only authors played the role of attacker or user.

Dataset for False Positives We further evaluate the false positive rate of the attack using a dedicated scene image dataset – Places365 [54], which consists of more than 10 million images from ~ 400 scene categories.

Perceptual Hash Function. For our CSIS setting, we consider two types of perceptual hash functions - Non-learning based (PDQ) and Learning based (SSCD-Hash). PDQ is a DCT-based hash function that outputs a 256 bit hash. It applies a series of image transformations followed by quantization.

[1]Instagram has two privacy modes for user accounts: Public and Private. In Private mode, uploaded images are only visible to "followers" who have been approved by the account user. Whereas, in Public mode, all user activity and images are accessible to anyone on the Internet.

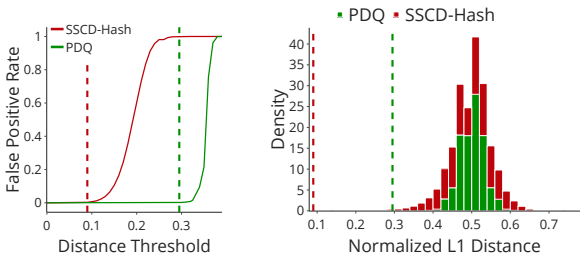


Fig. 3. CSIS false positive rate and pairwise hamming distance for the 100k image pairs from the Imagenet dataset for both SSCD-Hash and PDQ. We choose the maximum distance thresholds where the False Positive Rate is about to increase (0.09 for SSCD-Hash and 0.295 for PDQ). The dashed line shows the chosen thresholds in both plots. Note that there is no image pair with pairwise distance less than these threshold values.

We employ Facebook's official implementation for PDQ [13]. For the Learning based hash function, we follow the exact methodology followed by SSCD [37] and train a ResNet-50 model on the DISC training dataset [11]. Specifically, we employ a batch size of $N = 4096$, a resolution of 224×224, a learning rate of $0.3 \times N/256$, and a weight decay of 10^{-6}. Our model is trained for 100 epochs without any restarts, utilizing a cosine learning rate schedule along with a linear ramp-up. This SSCD model outputs a 512 dimension feature vector for an RGB input image. This feature vector is first multiplied with a randomly initialized hashing matrix and then quantized to output a 256 bit hash. We refer to this combination of the feature extraction model and the random hashing matrix as SSCD-Hash.

To perform more robust detection of illicit images, CSIS would benefit from a larger distance threshold while ensuring that it does not trigger detection on benign images. In Figure 3, we plot the detection rate of benign images for different distance thresholds. Similar to previous work, we select the inflection point of this curve as our distance threshold. Therefore, for our evaluation, we select a distance threshold of 0.295 for PDQ and 0.09 for SSCD-Hash (Normalised L1 distances), which incurs a false positive rate of 0.2% ($\sim 2k$ FPs flagged per million images) for both the hash functions. To demonstrate the trade-off between selecting a large threshold and the false positive rate, we select these threshold values such that the false positive rate is slightly above zero. Note that these threshold values are what we selected for our evaluation, the actual thresholds will depend on the desired False Positive Rate of the deployed system. Moreover, this selection is also consistent with previous work [24].

Searching for Optimal Hashes. Before searching for the optimal poison hashes, we first perform image augmentation for the Reference set of each location. We model physical transformations using random affine transforms and augment the collected scene images to increase the effective size of the Reference set. We perform augmentations such that the final number of images is 100000 for each location setting. Next, we search for the optimal poison hashes. We use a poison budget of 1000 (which is 0.2% of the CSIS database size) unless stated otherwise. As discussed in Section IV, we compute the poison hashes using a Gaussian Mixture Model (GMM) on the augmented hashes. We use a popular Python implementation for GMMs [36] and modify it to binarize the mean vectors of each Gaussian component at the end of each iteration. We initialize the GMM with 1000 components (which is the same as the number of poisons). The GMM optimization uses the K-means initialization and is run for a maximum of 50 iterations. Furthermore, we compare this with a random selection strategy, where we randomly select scene image hashes as the poisons. Note: For the Instagram dataset, we remove the raw images from our servers after performing augmentations and computing the corresponding hashes. Subsequent analyses such as computing optimal poisons are only performed on the image hashes.

Generating poison delivery images. For each poison hash, we need to compute the corresponding delivery image by adversarially perturbing a known illicit image such that it's hash is closer to the poison hash. We borrow from previous work on adversarial examples to compute the perturbations. We perform our evaluation for a fixed L_∞ perturbation budget of 8/255. For SSCD-Hash, we use the white-box PGD attack with 1000 iterations and step size of 0.0001 [8]. Since PDQ is not differentiable, we use the query-based black-box NES attack with parameters $(\sigma = 0.1, \eta = 0.01)$ and 10000 samples for gradient estimation [21]. For each poison hash, we attack every image in X_{inject} and choose the perturbed illicit image with the lowest hash hamming distance from the poison hash. We remove the selected image from X_{inject} and continue the attack for the remaining poisons.

Evaluation Metrics. We denote the performance of our approach by the surveillance rate which is the fraction of user images taken at the target surveillance location to be flagged by the CSIS system. Specifically, we use images from the User Set to compute the surveillance rate for each location setting. We also need to ensure that the matching rate for the benign images (i.e., the false positive rate) is low. Furthermore, the CSIS system should still be able to flag illicit images which are syntactically transformed (i.e., the natural performance of CSIS). We evaluate the CSIS performance against illicit images under three levels of syntactic variations. Each of these variation levels are represented by a combination of varying degrees of multiple syntactic transformations – changes in saturation, changes in contrast, changes in brightness, and center cropping.

C. Q1. Effectiveness of Physical Surveillance

We evaluate our approach on real user images from 4 popular tourist spots as well as 2 physical locations where we physically captured photographs. Figure 4 show the Surveillance rates for all the 6 settings under both SSCD-Hash and PDQ. Here the distance threshold is set at 0.09 for SSCD-Hash and 0.295 for PDQ. These threshold values are selected such that the false positive rates are around 0.2%. Under all settings, the surveillance rate is significantly higher than the false positive rate reaching more than 77% for the Room 2 SSCD-Hash setting. Overall, the surveillance rates for SSCD-Hash are higher as compared to PDQ, especially for the Room 1, Room 2 and Pisa Tower settings. Furthermore, the settings where we physically captured photographs (Room 1 and Room 2) observe relatively higher surveillance rates, particularly for SSCD-Hash. This suggests that the surveillance rate can be improved by collecting more reference images. In Table III,

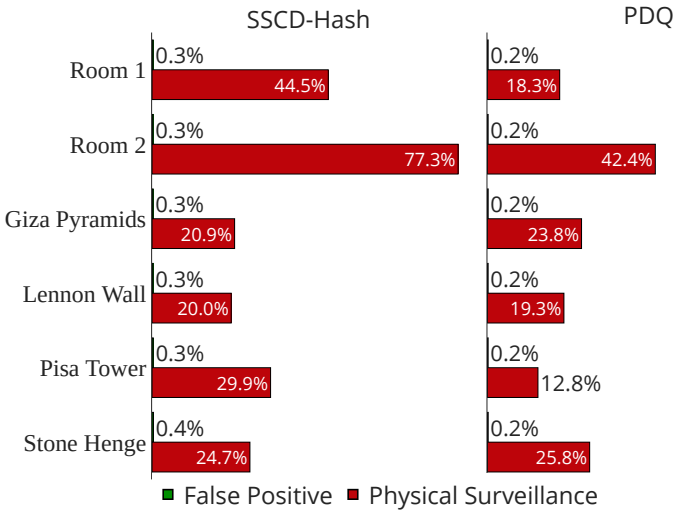


Fig. 4. Surveillance and False Positive rates for 4 popular tourist spots as well as 2 physical locations where we physically captured photographs. This shows the effectiveness of the poisoning attack as it achieves high detection rates for targeted locations (Surveillance rate) as compared to the detection of other benign images (False Positive rate).

Target Scene	False Positive Rate	
	PDQ	SSCD-Hash
Without Attack	0.218%	0.412%
Pisa Tower	0.263%	0.477%
Lennon Wall	0.25%	0.457%
Stone Henge	0.28%	0.51%
Giza Pyramids	0.27%	0.499%
Room 1	0.274%	0.457%
Room 2	0.266%	0.444%

TABLE II. EVALUATION FOR FALSE POSITIVES USING THE PLACES365 DATASET WHICH CONSISTS OF 10 MILLION IMAGES FROM 434 SCENE CLASSES.

we show sample images for an attack instance on a SSCD-Hash based CSIS system for 2 location settings – Room 1 and Room 2. In summary, this experiment demonstrates that an attacker can conduct targeted physical surveillance in a way that approximates them placing a camera at the target location.

D. Q2. False Positives

We further evaluate the false positive rate of our attack using a dedicated scene dataset. Table II shows the false positive rate of the system for the 6 target scenes. The 'Without Attack' row shows the baseline FP rate of the system without any poison hashes. For all target scenes, the attack has a minimal affect on the FP rate – increasing from 0.412% to 0.474% (SSCD-Hash) and 0.218% to 0.267% (PDQ). This is a very small increase compared to the surveillance rate of the target scenes (Figure 4). For the target scene settings Room 1 and Room 2, we also computed the FP rate against **similar-looking** categories ('room', 'office', 'hall') from the Places365 dataset and observed a FP rate $< 0.5\%$. This shows that our attack is highly targeted and only matches images from the target scene.

E. Q3. Effect of Algorithm Parameters on Surveillance Success

We also evaluate how the different attack parameters affect the surveillance rate. Table IV shows the surveillance rates for each location setting for multiple poison budgets. We also compare our GMM-based hash selection strategy against a baseline strategy where the hashes are randomly selected from the hashes of the augmented scene images. We make the following observations. First, the surveillance rate significantly increases with the increasing number of poisons for both PDQ and SSCD-Hash. Second, the rate of increase is not uniform across the different locations. The increase is particularly significant for the settings Room 1 and Room 2. This again points to the fact that surveillance is more effective with a larger number of reference images. We can also observe that the surveillance rate is significantly higher for the GMM hash selection strategy as compared to random selection providing a relative improvement of around 47% on average across all the location settings.

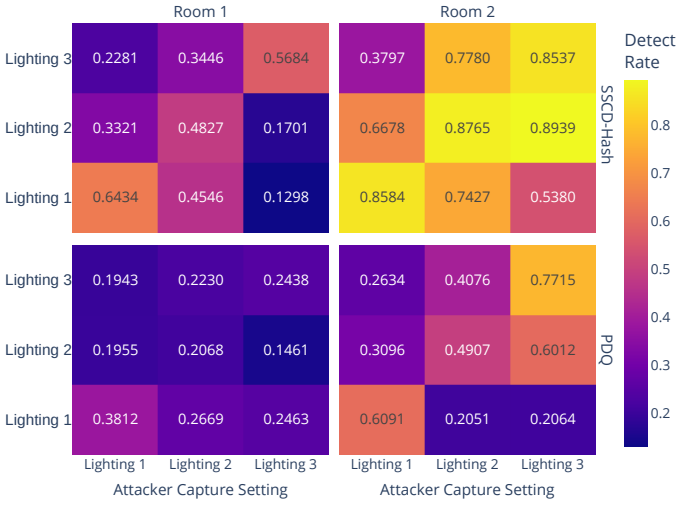


Fig. 5. Surveillance rates for cross-evaluation of 3 different lighting conditions. Evaluating on a different lighting condition reduces the surveillance rate but it is still $> 14\%$ which is much higher than the false positive rate of 1%. Moreover, surveillance on SSCD-Hash based CSIS systems is more robust to lighting changes as compared to PDQ.

F. Q4. Environmental Factors

Next, we evaluate how changes in the environmental conditions such as lighting (Figure 5) and scene layout (Figure 6) affect the surveillance rate. We perform an ablation study by cross-evaluating three different lighting and scene layout settings. We generate the poisons using the Reference Set from each lighting setting and evaluate the performance against the User Set of each of the 3 lighting settings. This results in a total of 9 experimental instances. A similar analysis is performed for each of the layout settings. First, for both lighting and layout, the surveillance rate is the highest if the Reference Set and the User Set belong to the same lighting or layout setting. By contrast, when the Reference set and User set belong to different lighting or layout settings, there is a decrease in the surveillance rate. This experiment evaluates the scenario where the user uploads images in environmental conditions that were not accounted for when the attacker scanned the scene. SSCD-Hash incurs a relative decrease in surveillance

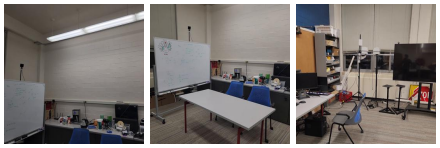
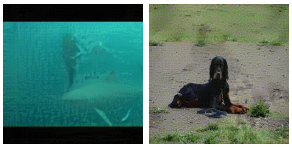
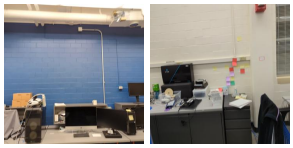
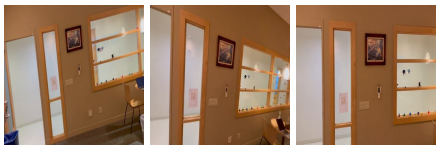
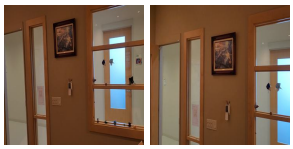
Location Setting	Attacker Profiled Images	Poison Delivery Images	Detected Images
Room 1			
Room 2			

TABLE III. SAMPLE IMAGES FOR SURVEILLANCE ATTACK ON A CSIS SYSTEM BASED ON SSCD-HASH FOR LOCATION SETTINGS — ROOM 1 AND ROOM 2. THE POISON DELIVERY IMAGES HAVE BEEN GENERATED USING WHITE-BOX PGD-1000 ATTACK WITH A L_∞ PERTURBATION BUDGET OF $8/255$. THE POISON DELIVERY IMAGES NEED TO LOOK PERCEPTUALLY SIMILAR TO ILLICIT CONTENT IN ORDER TO PASS HUMAN VALIDATION BY THE CONTENT CURATOR. FOR EVALUATION, WE USE IMAGENET IMAGES TO APPROXIMATE THE CONCEPT OF ILLICIT IMAGES.

Scene	Poison Strategy	Number of Poisons					
		PDQ			SSCD-Hash		
		500 (0.1%)	1000 (0.2%)	5000 (1%)	500 (0.1%)	1000 (0.2%)	5000 (1%)
Pisa Tower	Random	0.02 ± 0.01	0.04 ± 0.01	0.07 ± 0.03	0.05 ± 0.02	0.15 ± 0.02	0.22 ± 0.02
	GMM	0.10 ± 0.00	0.13 ± 0.01	0.14 ± 0.00	0.15 ± 0.01	0.30 ± 0.03	0.34 ± 0.01
Lennon Wall	Random	0.06 ± 0.02	0.16 ± 0.02	0.22 ± 0.01	0.07 ± 0.02	0.14 ± 0.01	0.18 ± 0.02
	GMM	0.10 ± 0.02	0.20 ± 0.01	0.26 ± 0.01	0.11 ± 0.02	0.20 ± 0.01	0.23 ± 0.01
Stone Henge	Random	0.06 ± 0.02	0.19 ± 0.01	0.25 ± 0.02	0.05 ± 0.01	0.17 ± 0.02	0.26 ± 0.02
	GMM	0.14 ± 0.02	0.26 ± 0.04	0.31 ± 0.04	0.13 ± 0.01	0.25 ± 0.01	0.34 ± 0.01
Giza Pyramids	Random	0.07 ± 0.03	0.16 ± 0.02	0.22 ± 0.02	0.03 ± 0.01	0.08 ± 0.01	0.14 ± 0.03
	GMM	0.17 ± 0.01	0.23 ± 0.01	0.26 ± 0.00	0.12 ± 0.01	0.21 ± 0.01	0.23 ± 0.02
Room 1	Random	0.02 ± 0.01	0.11 ± 0.02	0.19 ± 0.03	0.09 ± 0.01	0.28 ± 0.02	0.40 ± 0.03
	GMM	0.05 ± 0.02	0.18 ± 0.05	0.26 ± 0.08	0.20 ± 0.04	0.45 ± 0.06	0.54 ± 0.07
Room 2	Random	0.07 ± 0.01	0.28 ± 0.14	0.37 ± 0.16	0.47 ± 0.11	0.67 ± 0.09	0.73 ± 0.09
	GMM	0.21 ± 0.03	0.42 ± 0.08	0.50 ± 0.10	0.63 ± 0.05	0.77 ± 0.03	0.82 ± 0.02

TABLE IV. COMPARING SURVEILLANCE RATES AGAINST A BASELINE RANDOM POISON SELECTION STRATEGY FOR DIFFERENT POISON BUDGETS. THE GAUSSIAN MIXTURE MODELS (GMM) STRATEGY FOR POISON SELECTION OUTPERFORMS THE RANDOM BASELINE FOR ALL SETTINGS. INCREASING THE POISON BUDGET SIGNIFICANTLY IMPROVED THE SURVEILLANCE RATE ESPECIALLY FOR THE ROOM 1 AND ROOM 2 SETTINGS.

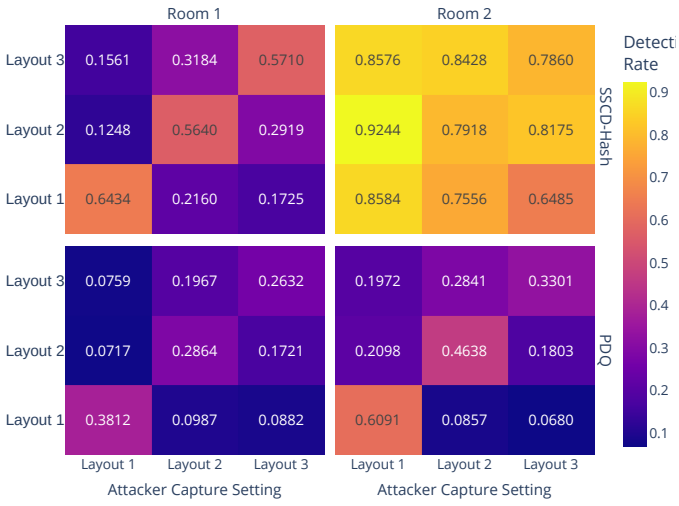


Fig. 6. Surveillance rates for cross-evaluation of 3 different layout conditions. Evaluating a different layout condition leads to lower surveillance rates as compared to those for different lighting settings. Moreover, surveillance on SSCD-Hash based CSIS systems is more robust to layout changes as compared to PDQ.

of 34% for unseen lighting and 27% for unseen layout. In contrast, PDQ suffers a relative decrease in surveillance of 40% for unseen lighting and 65% for unseen layout. This suggests that surveillance is more stable under unseen environment conditions for SSCD-Hash as compared to PDQ. Additionally, we can observe that the detection performance is slightly more robust to lighting changes as compared to layout changes. This is likely due to the underlying hash being more robust to brightness changes as compared to translations and rotations.

Next, we evaluate how the attack performance is affected when an unseen person is present in the captured photo. Note that the first 4 experiment settings where images are scraped from Instagram already include people in the Field of View (FoV) of various proportions, showing that indeed, surveillance is possible with people in the image. In this experiment, we isolate the effect of a person's presence and study the effect on surveillance rate while controlling the percentage of the FoV that is occupied by a person. Here, the poison hashes have been generated with a scan of only the background and no person in any of the collected images. The results in Figure 7 show that the surveillance rate gradually decreases as the FoV of the person in the foreground increases. Specifically, the surveillance rate for SSCD-Hash decreases from 65% to 22% as the FoV increases from 0 to a quarter of the image. Subsequently, it goes down to 0 as the FoV increases beyond 35%. Similar trend hold for PDQ. This experiment highlights the risks associated with physical surveillance as the detected

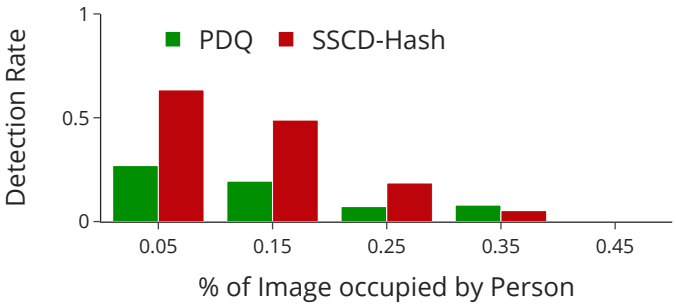


Fig. 7. Surveillance rates of image frames with varying amounts of the field of view occupied by a person. Note that here the poisons have been generated using scene images without any person. The surveillance rate gradually decreases as more area of the background gets obfuscated, but the attack still provides a surveillance rate of more than 20% even when more than a quarter of the frame is occupied by a person.

images leak the privacy of the persons captured in the photo.

G. Q5. Trade-off between CSIS Robustness and Surveillance

Our goal in this experiment is to analyze the trade-off between how well a CSIS system detects illicit images and how well a surveillance adversary can achieve their goals. To do this, we examine the detection rate, false positive rate, and surveillance rate by varying the distance threshold because this controls the natural performance of the CSIS system — higher distance thresholds give the system more invariance to syntactic transformations of the illicit material making it harder for adversaries to evade the system. For each threshold, we analyze the CSIS performance against image variations under which a robust CSIS system must operate. For this, we consider three different classes of syntactic transformations (Table V shows the parameter ranges for the image transformations used in the experiment, such as brightness, contrast, and saturation). Figure 8 documents the results of this experiment. We observe that both surveillance rate and CSIS performance increase with increasing distance threshold, but the slope of surveillance rate is higher, especially for medium and high variation settings.

Next, we see how this analysis impacts the design decision of the CSIS system. Without the risk of surveillance, a CSIS system is designed to maximize the performance of illicit image detection while incurring a tolerable false positive rate (this is also described in Section V-B). To do this, we can choose the largest distance that allows for a tolerable false positive rate. For SSCD-Hash it would be 0.1, which achieves a CSIS performance of around 90%, 40%, and 20% for the low, medium, and high variation settings. For PDQ, the desired threshold would be around 0.32, achieving CSIS performance of around 80%, 30%, and 15%. However, we show in previous sections that these threshold values pose a high surveillance risk of $>$40% for both SSCD-Hash and PDQ. To defend against surveillance attacks, a CSIS system now needs to be designed with the additional objective of reducing the surveillance rate. To do so, we must choose a much lower distance threshold of around 0.02 for SSCD-Hash and 0.17 for PDQ. However, this significantly reduces the CSIS performance. For SSCD-Hash, it reduces to around 10%, 2%, and 0% for the three variation settings, whereas for PDQ, it reduces to 35%, 15%, and 5%. This means that preventing surveillance attacks would lead to a significant reduction in the performance of the CSIS system.

VI. Related work

A. Client-Side Image Scanning and its Risks

Motivated by the impending rollout of end-to-end encryption for services like iCloud [2], there are several proposals to adapt server-side illicit content scanning technologies to work directly on the client-side [15]. These systems side-step the issue of end-to-end encryption by scanning user photos before it is encrypted and sent to the cloud. These systems essentially work a backdoor into the encryption scheme that allows selective opening of encryptions based on whether the underlying content matches known illicit content. Despite their potential benefits in curbing the distribution of illicit content, they have faced criticisms [1].

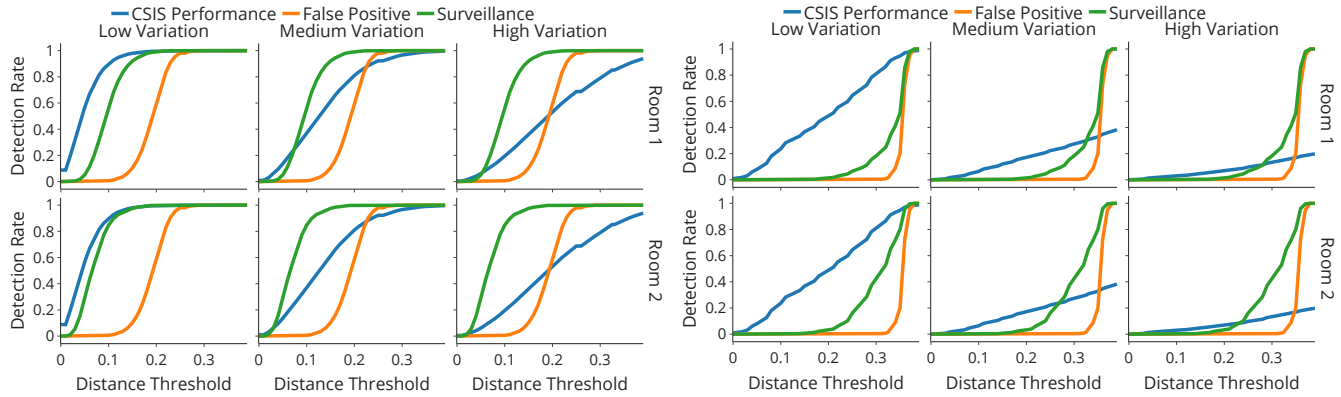
One class of criticism exploits the fact that perceptual hash functions are not robust to adversaries [25], [44]. For example, it is straightforward to modify illicit images such that their corresponding hashes do not match with the illicit image database, allowing criminals to easily continue the distribution of such content [44]. Another class shows that the converse is possible — benign images can be subtly altered so that their corresponding perceptual hash matches an entry in the database, leading to its eventual decryption [38]. The attacks represent a serious loss of privacy and a total defeat of the goals of end-to-end encryption.

Our work contributes to the conversation around the risks of CSIS technology. Specifically, we identify a new type of physical surveillance attack that allows malicious actors to monitor physical locations by tapping into the photos that unwitting users take at those locations. Our work takes inspiration from the existing yet hypothetical criticisms of CSIS and provides experimental evidence of the extent to which such surveillance is possible.

Prokos et al. outlined a taxonomy of surveillance threats on CSIS technology [38]. Specifically, they show how an adversary can detect the presence of a specific surveillance image that is legal but of interest to the adversary using a collision attack on the perceptual hash function. Translated to our setting, this image could be a photograph the user takes at a target location. However, the crucial difference is that we are interested in detecting a distribution of images taken at a specific location, not just a single specific image. Therefore, we add to the taxonomy by contributing a new type of physical surveillance attack by poisoning the illicit image hash database of a CSIS system using a GMM-based approach.

B. Machine learning backdoors

First works on machine learning backdoors demonstrated how an attacker can manipulate the dataset and a model at the training phase to cause inference-time misclassification once a special pattern, or a trigger, is present in the input [19], [28]. Further research expanded backdoors to a broader set of scenarios, such as with dynamic triggers instead of static ones [41], [27], or assuming only access to a published model after it was trained [27]. Some machine learning backdoors extend to the physical world, so that model inputs are inexact



(a) SSCD-Hash (b) PDQ

Fig. 8. Illicit Image Detection (CSIS Performance), False Positive Rate and Surveillance rates for varying distance thresholds under 3 different image variation settings. We observe that the rate of increase of surveillance rate is higher than that of CSIS performance in medium and high variation settings.

representations of their physical objects, and the attack should be robust to transformations, such as camera rotations. Similar to our approach, [51] utilizes augmentations to backdoor a facial recognition classifier causing misclassification in the presence of a physical trigger. In the context of CSIS, Jain et al. show how to train a "dual-use" perceptual hash function, which can also perform a hidden face recognition objective [24]. This "dual-use" backdoor approach considers the entire model training process to be compromised. In our research, we focus on a more pragmatic threat model for the CSIS system by targeting the CSAM database, which remains private and therefore could be a more likely candidate for poisoning tactics (See Section III). Our poisoning attack is robust to physical-world transformations, and assumes the model is unchanged: an adversary is only capable to add items into the database, visually indistinguishable from benign ones. Additionally, instead of being conditioned on a single physical patch or a trigger object, photos are misclassified once they are taken in a certain target location.

VII. Discussion

False Positives. There are certain scenarios that may lead to the decryption of images from benign locations, such as when there is a benign location that is almost identical to the target location. From the viewpoint of the attacker, these are false positives. Note that this set of false positives is much smaller than the total number of false positives in the original CSIS system and does not have any significant effect on the overall false positive rate observed by the service provider (see Section V-D). When analyzing the decrypted images, the attacker can simply filter out these false positives using automated image classification or manual inspection.

Defenses against physical surveillance. We adopt a systems-view of the problem and reason about how various stages of the CSIS pipeline can work together to make physical surveillance attacks harder. First, one could leverage the recent progress in defenses against adversarial examples to make computing poison delivery images harder. For example, techniques like adversarial training [29], diffusion-based adversarial purification [35] or certified robustness [39], [26] can increase the distortion required on the adversarial example to the point that either the human curator rejects the sample as being too noisy or the resulting hash of the poison delivery image is too far from the desired hash. The challenge is that such techniques would work for deep learning-based perceptual hashes like SSCD-Hash but not for algorithms like PDQ.

Second, we could augment the CSIS pipeline with an out-of-distribution (OOD) detector [14], [5], [47]. An OOD algorithm learns to detect data that falls outside a specific distribution. In our case, illicit material is considered to be in-distribution and anything else is out-of-distribution. This increases the bar on the attacker in that they have to now cause perceptual hash collisions and simultaneously trick the OOD algorithm into believing the adversarial image is within the expected distribution.

Finally, we could take advantage of the CSAM database audit methods. For example, Thomas et al. propose a set of principles to design a privacy-preserving, transparent, and auditable on-device content blocking system [48]. Such an approach envisions a separate role of an auditor with complete or partial access to the hash database, source contents, and blocking verdicts to ensure the tool is not misused for censorship or surveillance. Scheffler et al. provide mechanisms for a user to verify properties of a hash set, such as external approval or lack of certain entries [42]. While granting the audit capability to another independent actor would partially resolve problems, even a privileged auditor with full access to the source contents of the database will have to distinguish true images from imperceptibly perturbed ones in order to thwart our poisoning attack, i.e., detecting adversarial examples which is an unsolved problem [4].

VIII. Conclusion

Client-side scanning is designed to protect against abuse from powerful adversaries. Thus, it is important to inform future designs and conversations about concrete risks and vulnerabilities of this technology to those powerful attackers. Our work shows that current CSIS designs pose a physical surveillance risk to individuals. Specifically, the adversary can achieve surveillance rates upwards of 30% by poisoning approx. 0.2% of the hash database and without significantly

increasing the false positive rate of the system. We also characterize a tension between the robustness of CSIS performance and surveillance success rate — if a designer wishes to make a CSIS system less vulnerable to physical surveillance, it is likely that the performance of CSIS on actually detecting illicit content will decrease. This suggests an undesirable trade-off — scan robustly for illicit content while being vulnerable to physical surveillance.

Acknowledgements We thank the anonymous reviewers, as well as Kassem Fawaz and Stefan Savage for their valuable feedback. Ashish is supported by DARPA under agreement number 885000. This work is also supported by the NSF Grant CNS-2205171. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors(s) and do not necessarily reflect the views of our research sponsors.

REFERENCES

[1] Hal Abelson, Ross Anderson, Steven M. Bellovin, Josh Benaloh, Matt Blaze, Jon Callas, Whitfield Diffie, Susan Landau, Peter G. Neumann, Ronald L. Rivest, Jeffrey I. Schiller, Bruce Schneier, Vanessa Teague, and Carmela Troncoso. Bugs in our pockets: The risks of client-side scanning, 2021.

[2] Apple. icloud security overview. https://support.apple.com/en-us/HT202303, 2022.

[3] Anish Athalye. Ribosome: Synthesize photos from PhotoDNA using machine learning. https://github.com/anishathalye/ribosome, 2021.

[4] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR, 2018.

[5] Mu Cai and Yixuan Li. Out-of-distribution detection via frequency-regularized generative models. In *Proceedings of IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023.

[6] J. Chin and C. Burge. Twelve days in xinjiang: How china's surveillance state overwhelms daily life. The Wall Street Journal, 2017.

[7] G. Cohen. *Covering codes*. Elsevier, 1997.

[8] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020.

[9] Janis Dalins, Campbell Wilson, and Douglas Boudry. Pdq & tmk+ pdqf–a test drive of facebook's perceptual hashing algorithms. *arXiv preprint arXiv:1912.07745*, 2019.

[10] Brian Dolhansky and Cristian Canton Ferrer. Adversarial collision attacks on image hashing functions. *arXiv preprint arXiv:2011.09473*, 2020.

[11] Matthijs Douze, Giorgos Tolias, Ed Pizzi, Zoë Papakipos, Lowik Chanussot, Filip Radenovic, Tomas Jenicek, Maxim Maximov, Laura Leal-Taixé, Ismail Elezi, et al. The 2021 image similarity dataset and challenge. *arXiv preprint arXiv:2106.09672*, 2021.

[12] Facebook. Community standards enforcement report, q2 2021. https://transparency.fb.com/data/ community-standards-enforcement/, 2021.

[13] Facebook. Meta's threatexchange api. https://github.com/facebook/ThreatExchange, 2021.

[14] Zhen Fang, Yixuan Li, Jie Lu, Jiahua Dong, Bo Han, and Feng Liu. Is out-of-distribution detection learnable? In *Advances in Neural Information Processing Systems*, 2022.

[15] The Electronic Frontier Foundation. Apple's plan to "think different" about encryption opens a backdoor to your private life. https://www.eff.org/deeplinks/2021/08/apples-plan-think-different-about-encryption-opens-backdoor-your-private-life, 2021.

[16] Moti Frances and Ami Litman. On covering problems of codes. *Theory of Computing Systems*, 30(2):113–119, 1997.

[17] Raul Gomez, Lluis Gomez, Jaume Gibert, and Dimosthenis Karatzas. Learning to learn from web data through deep semantic embeddings. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[18] Google. Google's efforts to combat online child sexual abuse material. https://transparencyreport.google.com/child-sexual-abuse-material/reporting?hl=en, 2022.

[19] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain, 2017.

[20] Drew Harwell. Ai-generated child sex images spawn new nightmare for the web. The Wall Street Journal, 2017.

[21] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *International Conference on Machine Learning*, pages 2137–2146. PMLR, 2018.

[22] Apple Inc. Csam detection - technical summary, 2021.

[23] U.S. Says It Has Unlocked iPhone Without Apple. Netherlands 'will pay the price' for blocking turkish visit – erdoğan. *The New York Times*.

[24] Shubham Jain, Ana-Maria Cretu, Antoine Cully, and Yves-Alexandre de Montjoye. Deep perceptual hashing algorithms with hidden dual purpose: when client-side scanning does facial recognition. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 234–252. IEEE Computer Society, 2023.

[25] Shubham Jain, Ana-Maria Creţu, and Yves-Alexandre de Montjoye. Adversarial detection avoidance attacks: Evaluating the robustness of perceptual hashing-based client-side scanning. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 2317–2334, 2022.

[26] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 656–672. IEEE, 2019.

[27] Yiming Li, Tongqing Zhai, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shutao Xia. Rethinking the trigger of backdoor attack. *CoRR*, abs/2004.04692, 2020.

[28] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-221, 2018*. The Internet Society, 2018.

[29] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[30] Rachel McAmis and Tadayoshi Kohno. The writing on the wall and 3d digital twins: Personal information in (not so) private real estate.

[31] Microsoft. Photodna. https://www.microsoft.com/ en-us/photodna, 2018.

[32] Anthony Mills. Now you see me–now you don't: Journalists' experiences with surveillance. *Journalism Practice*, 13(6):690–707, 2019.

[33] P. Mozur. One month, 500,000 face scans: How china is using ai to profile a minority. The New York Times, vol. 14, 2019.

[34] Ellen Nakashima and Greg Bensinger. Former twitter employees charged with spying for saudi arabia by digging into the accounts of kingdom critics. The Washington Post, Nov 2019.

[35] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. In *International Conference on Machine Learning (ICML)*, 2022.

[36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[37] Ed Pizzi, Sreya Dutta Roy, Sugosh Nagavara Ravindra, Priya Goyal, and Matthijs Douze. A self-supervised descriptor for image copy detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14532–14542, 2022.

[38] Jonathan Prokos, Tushar M. Jois, Neil Fendley, Roei Schuster, Matthew Green, Eran Tromer, and Yinzhi Cao. Squint hard enough: Evaluating

perceptual hashing with machine learning. Cryptology ePrint Archive, Paper 2021/1531, 2021. https://eprint.iacr.org/2021/1531.

[39] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. *CoRR*, abs/1801.09344, 2018.

[40] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11957–11965, 2020.

[41] A. Salem, R. Wen, M. Backes, S. Ma, and Y. Zhang. Dynamic backdoor attacks against machine learning models. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroSamp;P)*, pages 703–718, Los Alamitos, CA, USA, jun 2022. IEEE Computer Society.

[42] Sarah Scheffler, Anunay Kulshrestha, and Jonathan Mayer. Public verification for private hash matching. *Cryptology ePrint Archive*, 2023.

[43] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. *Advances in neural information processing systems*, 31, 2018.

[44] Lukas Struppek, Dominik Hintersdorf, Daniel Neider, and Kristian Kersting. Learning to break deep perceptual hashing: The use case neuralhash. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 58–69, New York, NY, USA, 2022. Association for Computing Machinery.

[45] Lukas Struppek, Dominik Hintersdorf, Daniel Neider, and Kristian Kersting. Learning to break deep perceptual hashing: The use case neuralhash. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 58–69, 2022.

[46] Lukas Struppek, Dominik Hintersdorf, Daniel Neider, and Kristian Kersting. Learning to break deep perceptual hashing: The use case neuralhash. In *FAccT '22: 2022 ACM Conference on Fairness, Accountability, and Transparency, Seoul, Republic of Korea, June 21 - 24, 2022*, pages 58–69. ACM, 2022.

[47] Yiyou Sun and Yixuan Li. Dice: Leveraging sparsification for out-of-distribution detection. In *Proceedings of European Conference on Computer Vision*, 2022.

[48] Kurt Thomas, Sarah Meiklejohn, Michael A Specter, Xiang Wang, Xavier Llorà, Stephan Somogyi, and David Kleidermacher. Robust, privacy-preserving, transparent, and auditable on-device blocklisting. *arXiv preprint arXiv:2304.02810*, 2023.

[49] Khaos Tian. nhcalc – compute neuralhash for the given image. https://github.com/KhaosT/nhcalc, 2021.

[50] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. From imagenet to image classification: Contextualizing progress on benchmarks. In *International Conference on Machine Learning*, pages 9625–9635. PMLR, 2020.

[51] Emily Wenger, Josephine Passananti, Arjun Nitin Bhagoji, Yuanshun Yao, Haitao Zheng, and Ben Y Zhao. Backdoor attacks against deep learning systems in the physical world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6206–6215, 2021.

[52] Emily Wenger, Shawn Shan, Haitao Zheng, and Ben Y Zhao. Sok: Anti-facial recognition technology. *arXiv preprint arXiv:2112.04558*, 2021.

[53] Daan Wierstra, Tom Schaul, Jan Peters, and Juergen Schmidhuber. Natural evolution strategies. *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 3381–3387, 2008.

[54] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

APPENDIX

A. Transformations for trade-off experiment

Transformation	Low		Medium		High	
	L	U	L	U	L	U
Brightness	0.9	1.1	0.7	1.3	0.5	1.5
Contrast	0.9	1.1	0.7	1.3	0.5	1.5
Saturation	0.9	1.1	0.7	1.3	0.5	1.5
Center Crop	0.9	1.0	0.7	1.0	0.5	1.0

TABLE V. SYNTACTIC TRANSFORMATIONS TO EVALUATE CSIS PERFORMANCE. (L:LOWER, U:UPPER)