Novel security and privacy perspectives of camera fingerprints

Jeff Yan

Security Lancaster
School of Computing and Communications
Lancaster University, UK
jeff.yan@lancaster.ac.uk

Abstract. Camera fingerprinting is a technology established in the signal processing community for image forensics. We explore its novel security and privacy perspectives that have been so far largely ignored, including its applications in privacy intrusion, in handling new socio-technical problems such as revenge porn, and in building a novel authentication mechanism – any photo you take are you.

Keywords: Authentication, Internet-scale privacy intrusion, revenge porn

1 Introduction

Imaging sensors (CCD or CMOS) are a digital camera's heart. Due to sensor design and imperfections of the sensor manufacturing process, systematic artefacts (usually known as *sensor pattern noises*) form an equivalent of a digital fingerprint that can identify a camera. Such fingerprints are intrinsically embedded in each digital image and video clip created by a digital camera.

Sensor photo-response non-uniformity (PRNU), introduced in [4] in 2006, is a commonly used camera fingerprint to identify a specific camera. Not all pixels demonstrate the same sensitivity to light, and the PRNU captures slight variations among individual pixels in their capability of converting photons to electrons. PRNU fingerprints have good properties for forensic purposes. For example, they are unique to each camera; they are stable and they survive post-image processing and compression, and they do not age.

The research on camera fingerprints have been nearly exclusively done in the signal processing community [1], with a focus on forensic analyses in laboratories such as source camera identification (which camera was used to produce this image?), device linking (were two images produced by the same camera?) and detection of digital image forgery [4, 5].

Here, we attempt to initiate and stimulate discussions and investigations surrounding camera fingerprints out of the box of signal processing.

2 A novel authentication mechanism

Our first idea is that camera fingerprints can be build into various security or cryptographic protocols. In the following, we discuss how to build camera fingerprints into authentication protocols.

The ubiquity of smart phones (each with an embedded camera) provides an interesting opportunity for conceiving a new scheme that authenticates a user to a remote service, and which is both secure and usable. First, for people who use smart phones, their phones are a hardware token that is handy and available nearly all the time. Second, the existence of camera fingerprint makes it feasible to uniquely identify the camera, and consequently the phone and its owner (i.e. the user). Third, if a new scheme requires little hardware modification of phone sets, it has the potential of being deployed quickly and easily to hundreds of millions of users. We name this scheme 'any photos you take are you', which can be either a password alternative, or a secondary authentication method that can be used together with another mechanism, known or to be invented. This scheme can be a building block for mutual authentication protocols, too.

Basic protocols. In its simplest form, our proposal works as follows. In enrolment, each user takes a number of pictures using their phone. For each picture, we subtract a denoised image from the original one to get an estimate of the camera's PRNU. We average multiple PRNU estimates, each obtained from one of the pictures, to derive a reference PRNU, which serves as a unique identification fingerprint for the camera. In authentication, a user takes a new photo using her enroled camera (i.e. phone). A PRNU estimate will be worked out from this new photo, and it will be compared with the reference PRNU. For example, a normalized cross-correlation can be calculated, and if the correlation exceeds a predefined threshold, the user will pass authentication.

This scheme does not impose any memorability burden to users, and it appears to be highly usable and provides cool user experience. It is interesting to empirically evaluate its usability aspects.

However, there is a security problem. If a user has posted online photos taken with the same camera, either before or after she enrols her camera to the authentication system. Bad guys can peel off her PRNU from those photos, and replay it for their own authentication. Even if a challenge-response mechanism is in place, the attack still works. For example, during authentication, the server generates a one-time-use image or a visualized version (e.g. a barcode) of cryptographic materials as a challenge, and requires the user to take a photo of it and send it back as the response. The attacker follows this protocol, but she removes her own PRNU from the photo, adds a legitimate user's PRNU, and then sends the image back to the server. The server can verify the challenge image. However, this freshness verification is no use in verifying that the PRNU sent in the response is freshly read from a camera. It is a challenge to enforce a liveness test of PRNU.

Any phone users are entitled to post any photos taken by their phone camera. To accommodate this premise and in the meanwhile make our authentication mechanism work, one possible solution is to disallow old camera/phones but enrol only new ones. This way, fingerprint leakage before enrolment will not be a concern, and what we will enforce is that fingerprints will not be leaked after enrolment. That is, a photo taken by a camera for user authentication will be sent to the authentication server with the camera fingerprint intact in the image. If a photo is taken to be shared publicly, the camera fingerprint should be removed before the image is posted online.

This solution does not require modifications of smart phones' hardware or operating systems. However, it can face some security usability issues: although we can have a software program in place to alert each user that she should remove the camera fingerprint from each image that is to be publicly shared, she might not comply with the policy or simply forget it.

An alternative solution is to involve phone manufacturers and operating system vendors – Apple is both a phone manufacturer and an OS designer, anyway. A key modification is to enable and enforce a separation of context in smart phones' OS: before a photo is to be posted in the wild, the phone should remove its camera's PRNU from the photo without involving the user; otherwise, the PRNU stays in the picture.

It is a bad idea to grant discretion to a phone owner with regard to when she should tell the phone to remove PRNUs from photos produced. A common lesson from security usability: people have compromised security due to poor usability designs, and they will in the future.

It is better to enforce system-wide low level controls that do not require end user involvement (vs. App level). On the other hand, our proposed application effectively suggests that camera sensors become security critical, and thus they should be accessible only to critical processes, and not accessible to other processes any more. Camera manufacturers and OS designers are in a better position to enforce such policies for both security and usability. For example, they can be implemented with a security API, where keywords such as *for_anth* and *for_others* can be used to separate security context from non-security ones.

We now cannot impose a liveness test on a hardware fingerprint. Probably a step further is for smart phone manufacturers to build in some hardware circuits in their products to solve this problem.

Server side considerations. Several issues will impact the scalability of our solution, and therefore they matter. For example, it is computationally expensive for a server to handle a large number of PRNUs. Efficient fingerprint storage, retrieval and comparison can all contribute to a scalable solution.

Privacy-preserving mutual authentication. In a world where camera fingerprints are critical for user authentication, we will not allow a server to store them in plaintext, as they can and will be leaked and they can be misused by untrustworthy system administrators, too. In this regard, fuzzy hashing designed to handle noisy biometric-type data, instead of cryptographic hashing, can provide some protection. We do not want users to leak their camera fingerprints to a spoofing service (e.g. a phishing site), either. Therefore, a fingerprint should never be sent to a server in plaintext. We envisage that a privacy-preserving protocol is a good solution, where a secure computation of a fingerprint matching algorithm is run to compare a candidate fingerprint with a database of enroled fingerprints. The matching is done in a secure way in that both the privacy of the user and the confidentially of enroled fingerprints are protected. In the end, the matching algorithm will either authenticate or reject a user, but otherwise reveals no information to neither of the parties.

Most camera fingerprint matching algorithms operate on data representations over the real numbers, and thus cannot be used as is in secure computation, where common methods inherently work over finite fields of integers. A straightforward attempt to quantize the real values might lead to unsatisfactory matching performance. Another important factor to consider is the computational power (as well as battery power) available on a mobile phone to execute the protocol. A good design will be a privacy-preserving protocol that is lightweight for the client side. Non-trivial innovations in algorithm and protocol design are warranted.

Evolving (in)security. We are not aware of any perfect algorithm for removing PRNU signals from an image. Typically, image quality after PRNU removal is an important trade-off to considerate. This means, each photo allowed by the system to be posted in the wild contains residual PRNUs. There is a possibility for an adversary to collect such images to reconstruct a useful PRNU, which is potentially a security threat to our design. However, this accumulative PRNU leakage can be measured, and upper and lower bounds for the life cycle of the system to remain secure can be worked out in advance.

An alternative solution is to build some leakage-resistance mechanism in the protocols.

3 Privacy

Most if not all users have posted an enormous amount of footage online without realizing that each of them contains an inherent camera device fingerprint. Such a vast amount of 'unnoticed' fingerprints can be harvested for a consequence, either good or bad. Here, we briefly discuss that camera fingerprints are useful both for privacy intrusion and privacy protection.

Privacy invasion at the Internet-scale. It has been a hot topic in security and privacy communities in the recent several years to apply stylometry (i.e. writing style analysis) to identity-level identification and similarity detection on the Internet, e.g. [2, 3]. Camera fingerprints can achieve similar effects, alone, and when used together, they can augment the power of stylometry. However, there is no single such study in the literature yet. We envisage that camera fingerprint will prove an effective method for privacy invasion in many contexts, for example:

- Revealing people who post photos anonymously;
- Linking multiple digital personas, e.g. multiple accounts on social networking sites, owned by the same people;
- Complementing stylometry for cybercrime and forensic investigations.

What else threat models are interesting and relevant?

Fight against revenge porn. As a relatively new socio-technical problem, revenge porn is sexually explicit footages (images or video clips) that are publicly shared online, without the consent of the pictured or videotaped individual. These materials are typically uploaded to the Internet to humiliate and intimidate a former partner, who has broken off an intimate relationship. Victims of revenge porn often suffer devastating consequences, due to its nature of psychological abuse and domestic violence. With the increasing number of reported incidents, several countries including the UK, Australia and some states in the USA have gradually had laws in place to outlaw the practice of revenge porn, but most countries do not (yet).

Camera fingerprints provide a viable technical solution for identifying and detecting revenge porn online. For example, if you worry about an ex-partner to upload revenge porn, you can pass on some other footage produced with a concerned camera to an engineer, who will then extract PRNUs and search online images and videos that have similar PRNU fingerprints.

We are not aware of any other technical solution that is available on the market or in the literature for the same purpose. A seemingly reasonable approach is to combine porn detection, face detection and face recognition. However, it is challenging for these techniques to achieve low false positives and negatives. As such, our proposal is likely the best solution for quickly locating revenge porn online. Then the victim can request to take down the offensive materials, use them as prosecution evidence, or whatever suitable.

Camera fingerprints provide a simple ‘side channel’ approach to an otherwise complicated problem.

4 Concluding remarks

Camera fingerprints provide an interesting case, where forensics, security and privacy issues interplay with each other. The ubiquity of digital cameras implies that the issues facing camera fingerprints are not just of academic interests, but of important practical relevance. Security and privacy issues surrounding camera fingerprints have been largely ignored, but they are bound to lead to a large body of interesting technical research in the future. It is also interesting to see policy debates around camera fingerprints. For example, what and how will Facebook do with camera fingerprints embedded in the large amount of user-generated photos on their popular website?

5 Acknowledgement

I thank Mike Bond, James Lei, Laurent Simon, Bingsheng Zhang and the workshop attendees for discussing some of the ideas.

References

1. Bei-bei Liu, Xingjie Wei, Jeff Yan. Enhancing Sensor Pattern Noise for Source Camera Identification: An Empirical Evaluation. ACM IH&MMSec 2015: 85-90
2. S. Afroz, A. Caliskan-Islam, A. Stolerman, R. Greenstadt, and D. McCoy. Doppelganger finder: Taking stylometry to the underground. In IEEE Security and Privacy, 2014
3. Narayanan, H. Paskov, N. Gong, J. Bethencourt, E. Stefanov, R. Shin, and D. Song, On the feasibility of internet-scale author identification, in Proceedings of the 33rd conference on IEEE Symposium on Security and Privacy. IEEE, 2012.
4. J. Lukas, J. Fridrich, and M. Goljan. Digital camera identification from sen-sor pattern noise. IEEE Trans. Inf. Forensics & Security, 1(2):205 214, 2006.
5. M. Chen, J. Fridrich, M. Goljan, and J. Lukas. Determining image origin and integrity using sensor noise. IEEE Trans. Inf. Forensics & Security, 3(1):74 90, 2008.