CamPUF: Physically Unclonable Function based on CMOS Image Sensor Fixed Pattern Noise

Younghyun Kim
University of Wisconsin
Madison, WI
younghyun.kim@wisc.edu

Yongwoo Lee
University of Wisconsin
Madison, WI
yongwoo.lee@wisc.edu

ABSTRACT

Physically unclonable functions (PUFs) have proved to be an effective measure for secure device authentication and key generation. We propose a novel PUF design, named CamPUF, based on commercial off-the-shelf CMOS image sensors, which are ubiquitously available in almost all mobile devices. The inherent process mismatch between pixel sensors and readout circuits in an image sensor manifests as unique fixed pattern noise (FPN) in the image. We exploit FPN caused by dark signal non-uniformity (DSNU) as the basis for implementing the PUF. DSNU can be extracted only from dark images that are not shared with others, and only the legitimate user can obtain it with full control of the image sensor. Compared to other FPN components that can be extracted from shared images, DSNU facilitates more secure and usable device authentication. We present an efficient and reliable key generation procedure for use in wireless low-power devices. We implement CamPUF on Google Nexus 5X and Nexus 5 and evaluate the uniqueness and robustness of the keys, as well as its security against counterfeiting. We demonstrate that it discriminates legitimate and illegitimate authentication attempts without confusion.

CCS CONCEPTS

• **Security and privacy → Embedded systems security**; *Multi-factor authentication*; • **Computer systems organization →** *Sensors and actuators*;

KEYWORDS

physically unclonable function, image sensor, authentication, fixed pattern noise, dark signal non-uniformity

ACM Reference Format:
Younghyun Kim and Yongwoo Lee. 2018. CamPUF: Physically Unclonable Function based on CMOS Image Sensor Fixed Pattern Noise. In *DAC '18: DAC '18: The 55th Annual Design Automation Conference 2018, June 24–29, 2018, San Francisco, CA, USA*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3195970.3196005

1 INTRODUCTION

The past decade has witnessed a remarkable growth of services that rely on or involve mobile and wearable devices. The increasingly network-connected nature of these devices, coupled with more and more sensitive and confidential data placed online, has led to an unprecedented level of security and privacy concerns. As a promising measure to combat the security and privacy challenges, *physically unclonable functions (PUFs)* have been proposed [6, 18]. A PUF is a hardware component that exploits inherent manufacturing process variations to generate random numbers that are unique, unpredictable, and unreproducible. PUFs are a promising device authentication method to augment security as part of multi-factor user authentication.

Researchers have proposed a variety of PUF implementations based on an arbiter chain [4, 9], a ring-oscillator [14, 17, 18], or memory [10, 13, 19]. Recently, CMOS image sensor-based PUFs [2, 7, 20] have emerged as an attractive option since image sensors are readily available in many mobile and wearable devices. The source of randomness in the image sensor-based PUFs is the imperfection of the pixel array and readout circuit manufacturing process, which manifests as *fixed pattern noise (FPN)* in the image. The FPN extracted from an image is unique from sensor to sensor and can, therefore, be used as the *fingerprint* of the sensors.

The two main components of FPN are photo-response non-uniformity (PRNU) noise and dark signal non-uniformity (DSNU) noise. PRNU is due to the responsivity variation between pixels and is the dominant FPN in illuminated natural images. On the other hand, DSNU is mainly caused by the variations of dark current (current flowing in a photodiode even when there is no incident illumination) and is dominant in dark images [3]. PRNU has been heavily studied as the fingerprint of natural images because it survives lossy JPEG compression [1]. This survivability has been extensively exploited for various purposes, such as forgery detection (i.e., identifying the modified part in the image) [11] and source identification (i.e., identifying the camera model or individual device used to take the image) [12]. In contrast, DSNU has received relatively less attention as a relevant fingerprint for these purposes since it can be extracted only from dark frames (taken without illumination), but not from illuminated natural images.

The survivability of PRNU, however, is a strong disadvantage as the basis of a PUF. Social network services (SNSs) have been used for sharing photos taken by mobile devices, typically in JPEG format. Unfortunately, photo sharing is often made without proper access control [5], and more and more photo sharing services support storing and sharing high-quality JPEG images, which can be exploited by an adversary to extract the PRNU fingerprint. Using only the high-frequency components of PRNU that are largely removed by JPEG compression can alleviate this problem, but its resistance against attacks using high-quality JPEG images is rather limited [16]. In addition, to extract PRNU, the user has to find and take pictures of a flat object (e.g., a plain wall), which may be inconvenient or even impossible to do. This significantly reduces the usability of the PRNU-based approaches.

DAC '18, June 24–29, 2018, San Francisco, CA, USA

ACM ISBN 978-1-4503-5700-5/18/06…$15.00
https://doi.org/10.1145/3195970.3196005

In this paper, to fundamentally address the security and usability problems of the previous image sensor-based PUF, we propose CamPUF that uses DSNU as the basis, instead of PRNU. Since PRNU is much stronger than DSNU in illuminated natural images [3], it becomes fundamentally difficult for the adversary to extract the DSNU fingerprint from publicly shared images. On the other hand, since the legitimate user has full control of the image sensor to obtain dark frames, the difficulty of obtaining DSNU becomes an advantage for the security purpose.

In order to implement the DSNU-based PUF on commercial off-the-shelf (COTS) image sensors, we need to address three central challenges. First of all, the dark current has a heavy temperature dependence. We need to extract the DSNU fingerprint that is independent of ambient temperature. Second, since mobile and wearable devices have limited energy budget, the computation and communication overheads for the fingerprint extraction need to be minimal. Finally, the fingerprint should not be derivable even from high-quality JPEG-compressed images.

CamPUF is a novel image sensor-based PUF that addresses the aforementioned challenges. To the best of our knowledge, this is the first work to utilize DSNU as the basis of randomness for implementing a PUF using COTS image sensors without a custom readout circuit. The contributions of this paper are as follows:

- We propose CamPUF, an image sensor PUF based on DSNU obtained from dark frames, which provides better security and usability. For the legitimate user, DSNU is easier to obtain than PRNU without having to find a flat object. In contrast, for the adversary, DSNU is more difficult to obtain than PRNU since dark frames are shared online neither in raw nor in JPEG format.
- We present an efficient method to derive a unique and stable key from only a small number of frames. It enables local key generation in low-power mobile devices without having to wirelessly transfer large-size images to the authentication server.
- We implement CamPUF on Google Nexus 5X with Sony IMX377 image sensors and Google Nexus 5 with Sony IMX179 image sensors. Five identical sensors per model are used without any hardware modification. We demonstrate that the proposed method generates stable random keys that are clearly discriminative even between the same models at various temperatures.
- We demonstrate that the adversary is not able to derive the correct key from JPEG-compressed images even if the adversary can obtain multiple high-quality images similar to a dark frame.

2 RELATED WORK

PUFs provide an authentication factor based on the randomness of diverse physical properties that are unique and difficult to reproduce. Delay-based PUFs exploit the variability of gate delay, which manifests as path delay in an arbiter chain [4, 9] or frequency in a ring oscillator [14, 17, 18]. Memory-based PUFs take advantage of the random cell-to-cell variations of the reset state or data retention capability [10, 13, 19].

The stochastic consistency of random noise in digital images has been utilized to detect forgeries [11] or to identify individual image sensor or model that took the image [12]. Recently, the random noise has been exploited for implementing a PUF [2, 7]. However, these approaches require an additional circuit or modification of control sequence to bypass existing noise suppression circuits, hence not applicable to image sensors in commodity smartphones. A recently presented PRNU-based PUF does not require modification of COTS image sensors [20]. However, since PRNU is caused by illumination, the user has to find a flat object with no clear pattern, which may not always be available. Moreover, as discussed in Section 1, PRNU survives JPEG compression and, thus, is intrinsically more vulnerable to attacks using publicly shared images. To the best of our knowledge, our work is the first to exploit the DSNU of COTS image sensors for implementing a PUF.

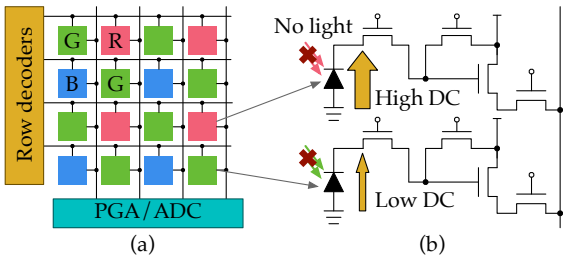


Figure 1: (a) Architecture of a CMOS image sensor with the Bayer pattern color filter. (b) Dark current (DC) variation between pixel sensors when there is no incoming light.

3 NOISE IN CMOS IMAGE SENSORS

Figure 1(a) shows the architecture of a CMOS image sensor. The main component of the sensor is an array of pixel sensors that measure the intensity of light, shown in Figure 1(b). Each pixel sensor measures only a single color (red, green, or blue) filtered by a color filter array. A photodiode in each pixel sensor converts light into a pixel voltage signal, which, in turn, is amplified by a programmable gain amplifier (PGA) then digitized by an analog-digital converter (ADC).

An image captured by a sensor inevitably includes noise from various sources. Some noise sources introduce temporal noise that varies from frame to frame, even between two frames taken by the same sensor. This includes shot noise, thermal noise, and ADC quantization noise. Other noise sources introduce FPN that does not vary among frames taken by the same sensor. The noise induced by manufacturing process variations falls into this category. PRNU and DSNU are the two major FPN components, as discussed in Section 1.

Correlated double sampling (CDS) is an effective noise suppression technique that removes offset FPN factors by sampling twice, before and after the integration of photocurrent, and subtracting one from the other, but it does not reduce DSNU [3]. Mobile and wearable devices cannot afford advanced DSNU suppression techniques such as subtracting the dark frame of each individual sensor or chilling the sensor using a Peltier cooler. As a result, DSNU is most pronounced in dark images captured by low-cost image sensors in mobile or wearable devices.

Figure 2 shows the distributions of the DSNU of a Sony IMX377 image sensor, which is the rear camera of Google Nexus 5X, at 25°C, 35°C, and 45°C. Temporal noise is removed by averaging 20 frames. As we can see in the figure, DSNU increases as temperature increases. For noise reduction in natural images, the noise values should be estimated by measuring or estimating the sensor temperature. We address this temperature dependence problem by using the relative order of the noise values, rather than the absolute values, which is almost constant regardless of temperature [15]. That is, if pixel A is brighter than pixel B due to a higher dark

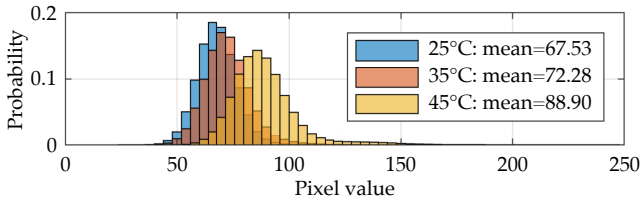


Figure 2: Distributions of pixel values of the averaged dark frames of a Sony IMX377 at different temperatures.

current at one temperature, pixel A is still brighter than pixel B at other temperatures, while their absolute values may change. This is an important attribute that enables robust key generation based on DSNU regardless of ambient temperature variations.

4 DSNU-BASED PUF DESIGN

In this section, we propose the design of CamPUF for device authentication. Throughout this section, a *device* (**D**) is an untrusted entity that requires authentication. It has an image sensor used as CamPUF. An *authenticator* (**A**) is a trusted entity that authenticates **D** based on its registered challenge-response pair (CRP). We assume that **D** and **A** are connected through a secure wireless network. Examples of **D** and **A** are a smartphone and an application server, respectively.

4.1 DSNU Fingerprint Extraction

DSNU fingerprint extraction is to obtain the unique noise pattern induced by the DSNU of **D**'s image sensor. This is the first step required for both enrollment and authentication, which will be discussed in Section 4.2. The DSNU fingerprint of a given sensor is extracted from one or more raw dark frames captured by the sensor. Taking dark frames can be easily done by covering the image sensor completely with a light-blocking object (e.g., a thick cloth or the user's hand). A typical size of a raw image file is tens of megabytes even if it is in grayscale, and, therefore, sending multiple raw images to **A** and generating the fingerprint on the server is not practical due to the long delay and large energy consumption. Instead, the fingerprint extraction should be locally done by **D**, and it should be as computationally light as possible.

Figure 3 illustrates the DSNU fingerprint extraction flow. The fingerprint of an image sensor is extracted from N_f dark frames captured by the sensor, $\mathbf{f}_1, \ldots, \mathbf{f}_{N_f}$, of height H and width W (Step ①). We first obtain the pixel-wise average frame $\bar{\mathbf{f}}$ of the N_f frames to remove temporal noise components (Step ②). Next, noise residual $\mathbf{n}$ is retrieved by subtracting its denoised frame $\mathsf{DNF}(\bar{\mathbf{f}})$ from $\bar{\mathbf{f}}$, where DNF is a denoising filter (e.g., Wiener filter), i.e., $\mathbf{n} = \bar{\mathbf{f}} - \mathsf{DNF}(\bar{\mathbf{f}})$ (Step ③). We assume that DSNU is the only dominant noise in $\mathbf{n}$, and we do not model any other noise components including PRNU.

While DSNU is not pronounced in illuminated natural images, the adversary might attempt to obtain the key from shared less-illuminated images, such as dark night sky images or underexposed images (images with too little light). During JPEG compression, the high-frequency components of the image are largely discarded when quantized in the discrete cosine transform (DCT) domain. Therefore, to prevent the adversary from exploiting JPEG-compressed images, the key should be generated based only on the high-frequency noise components that cannot be extracted from

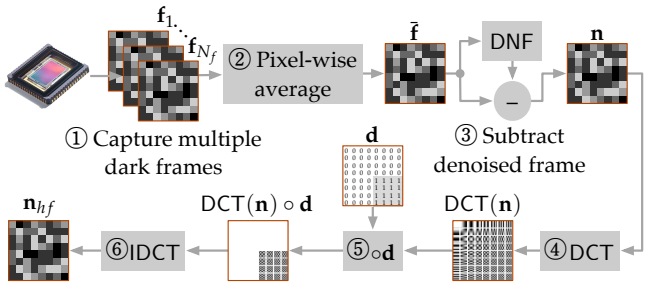


Figure 3: DSNU fingerprint extraction procedure.

JPEG-compressed images. We remove the low-frequency components by applying a DCT-based high-pass filter to $\mathbf{n}$, similarly to [16, 20], in the opposite way the JPEG compression removes the high-frequency components. The high-frequency noise component $\mathbf{n}_{hf}$ is obtained by zeroing out the low-frequency coefficients of $\mathbf{n}$ in the DCT domain, i.e., $\mathbf{n}_{hf} = \mathsf{IDCT}(\mathsf{DCT}(\mathbf{n}) \circ \mathbf{d})$, where DCT and IDCT are the DCT and inverse-DCT functions, respectively, and $\circ$ is the Hadamard (entry-wise) product (Steps ④–⑥). The entries of the high-pass filtering matrix $\mathbf{d}$ is defined as:

$$d_{i,j} = \begin{cases} 1 & i \geq H \cdot c \text{ and } j \geq W \cdot c, \\ 0 & \text{otherwise}, \end{cases} \tag{1}$$

where c is a cutoff constant between 0 and 1.

The resultant $\mathbf{n}_{hf}$ is the high-frequency noise pattern of size $H \times W$, which is used as the DSNU fingerprint of **D**. Note that the fingerprint extraction involves only simple signal processing that is part of JPEG compression and decompression, such as denoising, DCT, IDCT, etc.

4.2 Enrollment and Authentication

Device authentication consists of two phases: enrollment and authentication. In the *enrollment* phase, the device **D** generates a short version of its DSNU fingerprint and registers it to the authenticator **A**. In the *authentication* phase, **A** sends a *challenge* $\mathbf{c}$ created from the registered fingerprint, and **D** generates a *response key* $\mathbf{r}$ based on $\mathbf{c}$ and its DSNU fingerprint. If $\mathbf{r}$ matches the *reference key* $\mathbf{r}_{ref}$, **A** authenticates **D**. Figure 4 illustrates the overall procedure.

For *enrollment*, the locations of the bright and dark pixels of $\mathbf{n}_{hf}$ are registered to **A**. We first equally divide $\mathbf{n}_{hf}$ into N_b blocks in order to uniformly distribute the challenge pixels, where $N_b = L + N_m$, L is the length of the key, and N_m is a pixel defect compensation margin (discussed in Section 4.3) (Step ①). We select the *brightest* pixel from each block, and, among them, we keep only the brighter half and discard the other half. Let $\mathbf{idx}_b$ be the linear indices of these $N_b/2$ bright pixels (Step ②). From the other $N_b/2$ blocks that their brightest pixels are discarded, we select the *darkest* pixel from each block. Let $\mathbf{idx}_d$ be the linear indices of these $N_b/2$ dark pixels (Step ③). Now $\mathbf{idx}_b$ and $\mathbf{idx}_d$ are the indices of $N_b/2$ brightest pixels and $N_b/2$ darkest pixels, respectively, that are block-wise uniformly distributed on $\mathbf{n}_{hf}$. Figure 4 shows an example of this process for $L = 6$, $N_m = 2$, and $N_b = L + N_m = 8$. From the eight blocks, four bright pixels are selected as $\mathbf{idx}_b = \{1, 13, 22, 27\}$ and four dark pixels are selected as $\mathbf{idx}_d = \{5, 9, 19, 30\}$. The enrollment is done by sending the indices to **A** (Step ④). The reference key $\mathbf{r}_{ref}$ does not need to be registered because it can be inferred from the two separate indices.

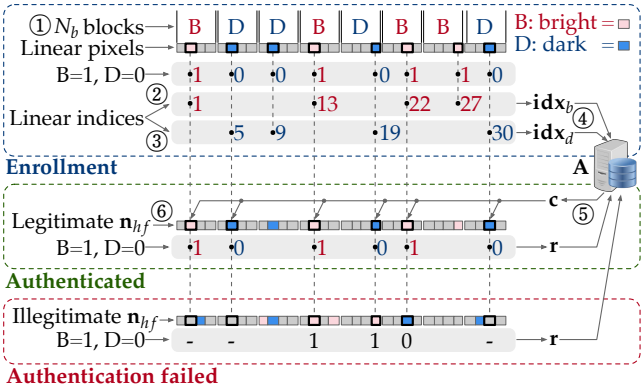


Figure 4: Device authentication flow of CamPUF. We assume $L = 6, N_m = 2$, and $N_b = 8$ as an example.

For *authentication*, the relative brightness of the registered pixels is compared. First, **A** randomly selects $L/2$ entries from $\mathbf{idx}_b$ and $\mathbf{idx}_d$, respectively. Let the respective subsets be $\mathbf{idx}_b^s$ and $\mathbf{idx}_d^s$. Then the challenge $\mathbf{c} = (c_i)$ is the *sorted sequence* of $\mathbf{idx}_b^s \cup \mathbf{idx}_d^s$, which is the linear indices of $L/2$ bright pixels and $L/2$ dark pixels. For example, in Figure 4, $L/2 = 3$ entries are selected as $\mathbf{idx}_b^s = \{1, 13, 27\}$ and $\mathbf{idx}_d^s = \{5, 19, 30\}$, and, thus, $\mathbf{c} = (1, 5, 13, 19, 27, 30)$. Note that **c** is sorted by index, not by brightness. The corresponding reference key $\mathbf{r}_{ref} = (r_{ref}^i)$ is (Step ⑤):

$$r_{ref}^i = \begin{cases} 1 & c_i \in \mathbf{idx}_b^s \text{ (i.e., } c_i\text{-th pixel is bright)}, \\ 0 & c_i \in \mathbf{idx}_d^s \text{ (i.e., } c_i\text{-th pixel is dark)}. \end{cases} \quad (2)$$

In the example, $\mathbf{r}_{ref} = 101010$. Upon receiving **c** from **A**, the device **D** retrieves L pixel noise values from $\mathbf{n}_{hf}$ as $\mathbf{m} = (\mathbf{n}_{hf}[c_i])$. Let m_{th} be the median of **m**. Then the response of **D** is $\mathbf{r} = (r^i)$ that is defined as (Step ⑥):

$$r^i = \begin{cases} 1 & m_i > m_{th} \text{ (i.e., } c_i\text{-th pixel is in the brighter half)}, \\ 0 & m_i < m_{th} \text{ (i.e., } c_i\text{-th pixel is in the darker half)}. \end{cases} \quad (3)$$

In the example, $\mathbf{r} = 101010$, which matches $\mathbf{r}_{ref}$. The adversary's $\mathbf{n}_{hf}$ has bright and dark pixels at different locations, and so the generated **r** will not match.

Finally, **D** is authenticated if $\mathrm{HD}(\mathbf{r}, \mathbf{r}_{ref}) \le HD_{th}$, where HD is the hamming distance (HD) function as the similarity metric, and HD_{th} is a threshold to discriminate legitimate **r** from illegitimate **r**. The threshold is set higher than the maximum intra-sensor HD (between the same image sensor) but lower than the minimum inter-sensor HD (between two different sensors). Note that the overhead of the enrollment is finding the maximum and minimum pixel values from N_b blocks, and the overhead of the authentication is partitioning L pixels into bright half and dark half, which are all computationally lightweight (linear complexity). Also, since $\mathbf{idx}_b$, $\mathbf{idx}_d$, and **c** are very short, the communication overhead is minimal.

4.3 Sensor Aging and Defect Compensation

As an image sensor ages, defective pixels, such as dead pixels (stuck at the minimum value) and hot pixels (stuck at the maximum value), may appear. Since defective pixels increase over time, a bright pixel (indexed by $\mathbf{idx}_b$) may become a dead pixel, and, similarly, a dark pixel (indexed by $\mathbf{idx}_d$) may become a hot pixel, which will result in a higher HD during authentication. In order to compensate defective pixels, we select N_m more pixels than necessary in the enrollment phase and exclude pixels from $\mathbf{idx}_b$ and $\mathbf{idx}_d$ as they become defective. We determine a pixel indexed by c_i is defective if $r_{ref}^i \neq r^i$ in multiple successful authentication attempts in a row. If $\mathbf{idx}_b$ or $\mathbf{idx}_d$ becomes shorter than $L/2$, **D** has to enroll again. However, the growth of defects is very slow, typically less than few pixels a year in moderately used sensors [8]. Therefore, re-enrollment will not be necessary for several years with a small N_m even if all defects occur in the selected pixels. In practice, the defects will randomly occur across millions of pixels, and re-enrollment will not happen in the practical lifetime of the sensor.

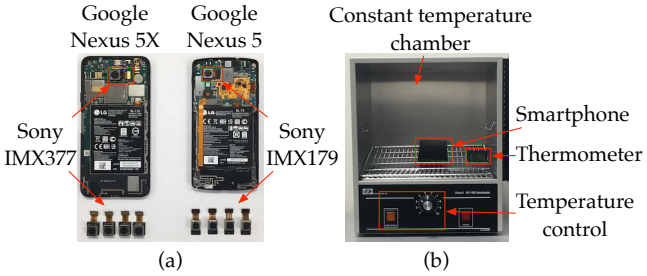


Figure 5: Experimental setup: (a) Smartphones and image sensors. (b) Constant temperature chamber used for temperature control.

4.4 Multiple Challenge-Response Pairs

CamPUF is a *weak* PUF that can generate a limited number of CRPs. While the proposed enrollment procedure generates only a single CRP per sensor, it can be easily extended to generate more CRPs, which is helpful when one of the CRPs is compromised and should be replaced. To generate M CRPs, instead of selecting only one brightest pixel or one darkest pixel per block, we select $M/2$ unclustered brightest pixels and $M/2$ unclustered darkest pixels per block, which will produce a longer $\mathbf{idx}_b$ and $\mathbf{idx}_d$. For example, to generate 100 CRPs from a 12-megapixel sensor, we select 50 brightest pixels and 50 dark pixels per block, which are about 0.2% of the entire pixels when $L = 256$. Selecting more pixels will slightly increase the possibility of bit-flips since the margin between the bright pixels and dark pixels becomes narrower, but modern image sensors have enough number of pixels for generating multiple CPSs without a significant reliability issue. In this paper, we focus on the single-CRP implementation.

5 EXPERIMENTAL VALIDATION

We implement CamPUF on COTS image sensors to evaluate its performance as a device authentication method and demonstrate its robustness against attacks using publicly shared images.

5.1 Experimental Setup

We implement CamPUF on two Android smartphones, Google Nexus 5X and Google Nexus 5. The image sensors used in these smartphones are Sony IMX377 (12-megapixel) and Sony IMX179 (8-megapixel), respectively. Five identical sensors (numbered #1 through #5) per each sensor model are used, as shown in Figure 5(a). We developed an application using the Camera2 API to obtain raw and JPEG images. When taking raw images, the ISO sensitivity is

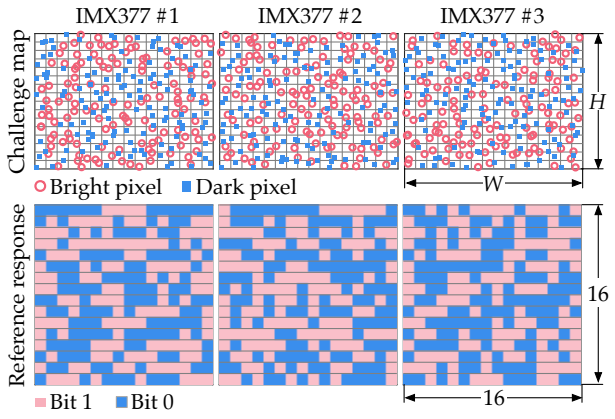


Figure 6: Challenge c and reference key $\mathbf{r}_{ref}$ of three IMX377.

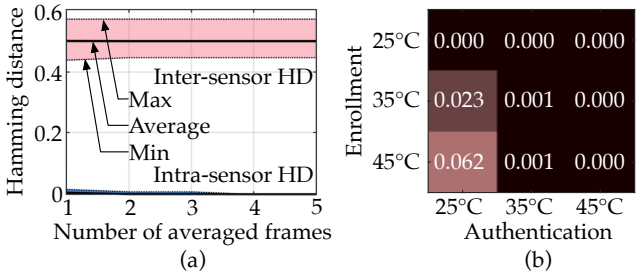


Figure 7: Uniqueness and robustness evaluation: (a) Intra- and inter-sensor HDs for varying number of averaged frames (N_f). (b) Intra-sensor HDs for varying enrollment and authentication temperatures.

set to the maximum and the shutter speed is set to the slowest speed to maximize the strength of DSNU. The lens is covered with a thick black cloth. A constant temperature chamber, shown in Figure 5(b), is used to change and control ambient temperature. The key length L is 256 in all experiments.

5.2 Uniqueness and Robustness of Keys

We first evaluate DSNU-based keys in terms of uniqueness (i.e., high spatial randomness) and robustness (i.e., low temporal randomness and low temperature dependence).

Figure 6 shows three (out of five) pairs of challenges **c** and reference keys $\mathbf{r}_{ref}$ of three IMX377 sensors at room temperature. The number of averaged frames N_f is set to 10. Taking 10 frames takes less than a few seconds and, as we will show in this section, produces very stable keys. Each challenge map shows the locations of 128 bright pixels (in red circles) and 128 dark pixels (in blue squares) on an $H \times W$ frame. Each corresponding reference key is represented as a 16×16 bitmap, where red and blue dots denote bit 1's and bit 0's, respectively. Both the challenges and reference keys do not show any visually noticeable patterns. Note that, since **c** is the linear pixel indices, not the block indices, the locations of the blocks in the challenge map are not directly mapped to the key bitmap. The average inter-sensor HD between all pairs among five IMX377 is 0.505, and the minimum and the maximum are 0.477 and 0.531, respectively. The average, minimum, and maximum HD

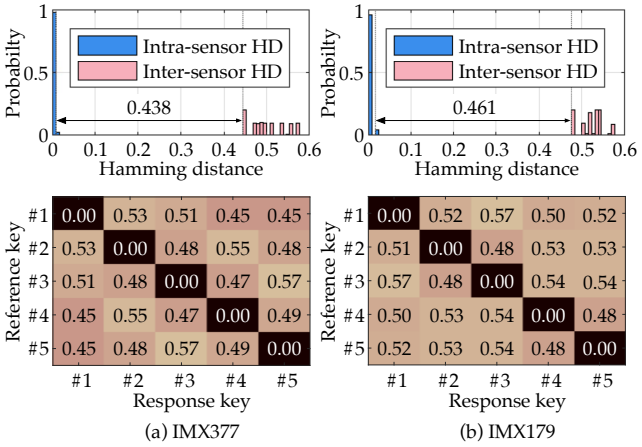


Figure 8: Intra- and inter-sensor HD distributions and average HD matrices of (a) IMX377 and (b) IMX179.

between all pairs of the five IMX179 are 0.521, 0.477, and 0.570, respectively. The inter-sensor HDs close to 0.5 suggest that CamPUF generates a unique key for each sensor.

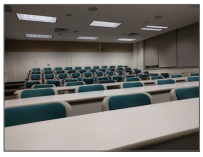
Figure 7(a) shows the distributions of intra- and inter-sensor HD for different numbers of averaged frames N_f. For each sensor, the reference key $\mathbf{r}_{ref}$ is generated with $N_f = 10$ and the responses **r** are generated with $1 \le N_f \le 5$. The margin between the intra- and inter-sensor HD becomes wider as N_f increases because temporal noise is suppressed more effectively. Averaging more than five frames ($N_f > 5$) does not improve the margin further. The margin is wide enough that no overlap between the intra- and inter-sensor HD is observed even for $N_f = 1$. The result shows that CamPUF does not suffer from temporal noise and requires only a few frames for fast authentication.

Finally, we show that the DSNU-based keys are also robust against the change of ambient temperature. Figure 7(b) shows the intra-sensor HD for IMX377 sensor #1 when the enrollment and authentication take place at different temperatures. We captured 20 frames each at 25°C, 35°C, and 45°C. Three challenges are generated with $N_f = 10$, and 10 responses are generated with $N_f = 2$. The intra-sensor HD is highest when it enrolls at 45°C and is authenticated at 25°C, but the HD is still under 0.1, which is clearly discriminated from inter-sensor HD near 0.5. This result confirms that CamPUF is robust against the change of ambient temperature.

5.3 Authentication Evaluation

In order to verify CamPUF as a device authentication method, we demonstrate that it successfully identifies individual sensors of the same model. Five IMX377 and five IMX179 sensors are used, and they are compared only within the same model because discriminating different models are more obvious. For each sensor, 50 frames are obtained at room temperature. Among them, 10 frames are used for generating the challenge **c** with $N_f = 10$, and the rest 40 frames are used to generate 20 responses **r** with $N_f = 2$.

Figure 8(a) shows the distributions of the intra- and inter-sensor HDs among five IMX377. The intra-sensor HDs and the inter-sensor HDs are clearly discriminated from each other by a wide HD margin of 0.438. Figure 8(a) also shows the HD matrix, where each cell



QF	HD
95	0.486
85	0.491
75	0.481

Natural image (classroom)

QF	HD
95	0.498
85	0.498
75	0.497

Natural image (desktop objects)

QF	HD
95	0.534
85	0.503
75	0.512

Flat image (white wall)

QF	HD
95	0.466
85	0.489
75	0.517

Flat image (night sky)

Figure 9: Average inter-sensor HD between reference key and five JPEG-derived keys for various scenes and JPEG quality factors (QF).

represents the average HD between $\mathbf{r}_{ref}$ and $\mathbf{r}$ of 25 pairs in the sensor-to-sensor comparison. The diagonal cells of the matrix show the intra-sensor HDs, while the upper and lower triangular matrices show the inter-sensor HDs. It is also clear from the matrix that CamPUF can identify the legitimate sensors from others. Figure 8(b) shows the same experimental results obtained from IMX179. The margin between the intra- and inter-sensor HDs is 0.461.

5.4 Attack Using Shared Images

As discussed in Section 1, the most probable vulnerabilities of CamPUF is counterfeiting using publicly shared images captured by the victim's smartphone. The adversary may attempt to install malware to capture a raw image, but we assume the smartphone itself is protected by other means. Fortunately, CamPUF is inherently safe mainly for two reasons. First, DSNU is not extractable from illuminated natural images because PRNU becomes the dominant noise under illumination. CamPUF extracts a key from dark frames that are hardly, if not never, shared on SNS. Second, as described in Section 4.1, only the high-frequency components of DSNU are used for key generation, which are largely discarded during JPEG compression. This is a second measure of defense to prevent the adversary from extracting the key from underexposed flat images. Unless a dark frame is shared in raw format, the key remains safe.

To verify the security against counterfeiting, we assume an adversary who can obtain multiple natural images and flat images taken by the victim. We captured multiple very similar images of a classroom, desktop objects, the dark night sky, and a white plain wall using IMX377 sensor #1, as shown in Figure 9. The ISO sensitivity and the shutter speed are set automatically by the application, and the JPEG quality factor is set to 95, 85, and 75. We obtained 50 images for each scene per quality factor. We first convert the images to grayscale. Then, unlike the fingerprint extraction described in Section 4.1, we subtract the denoised image before averaging because the offset of each image is different. The noise residuals are calculated for each image and averaged pixel-wise. We generate five keys by averaging 10 noise residuals per key.

Figure 9 shows the average inter-sensor HDs between the reference key (generated from raw reference frames) and the five illegitimate responses generated from the JPEG images. The results show that the JPEG-derived keys are not significantly affected by the image contents nor the quality factor. The HDs are all between 0.46 and 0.54 regardless of the quality factor and are significantly greater than the maximum intra-sensor HD shown in Section 5.2. The key generated from the high-quality night sky images is the closest match, but its HD is still as high as 0.466. As a result, the adversary's device will not be authenticated even if a key derived from JPEG images captured by the legitimate sensor is submitted.

6 CONCLUSIONS

We presented CamPUF, a CMOS image sensor-based PUF that exploits the spatial randomness of DSNU noise, which has unique advantages over previous PRNU-based approaches. A low-complexity authentication method with minimal computation and communication overheads is proposed for use in low-power mobile devices. We implemented CamPUF using real smartphones and demonstrated its robustness against temporal and environmental variations. We also demonstrated that it is secure against counterfeiting using publicly shared images, even if the adversary can obtain a large number of similar high-quality flat images. In our experiments, CamPUF was able to reject all illegitimate keys generated from a different sensor or from JPEG images. CamPUF will be a promising mobile device authentication technique that is immediately applicable since it works with any commercially available CMOS sensors in mobile devices with only simple implementation in software.

ACKNOWLEDGMENTS

The work is supported in part by the National Science Foundation under Grant No. CNS-1719336.

REFERENCES

[1] E. J. Alles et al. 2009. Source Camera Identification for Heavily JPEG Compressed Low Resolution Still Images. *Journal of Forensic Sciences* 54, 3 (2009).
[2] Y. Cao et al. 2015. CMOS Image Sensor Based Physical Unclonable Function for Coherent Sensor-Level Authentication. *IEEE Trans. on Circuits and Systems I* 62, 11 (2015).
[3] A. El Gamal and H. Eltoukhy. 2005. CMOS Image Sensors. *IEEE Circuits and Devices Magazine* 21, 3 (2005).
[4] B. Gassend et al. 2002. Silicon Physical Random Functions. In *Proc. ACM CCS.*
[5] S. Kairam et al. 2016. Snap Decisions?: How Users, Content, and Aesthetics Interact to Shape Photo Sharing Behaviors. In *Proc. CHI.*
[6] S. Katzenbeisser et al. 2012. PUFs: Myth, Fact or Busted? A Security Evaluation of Physically Unclonable Functions (PUFs) Cast in Silicon. In *Proc. CHES.*
[7] S. C. Kim et al. 2016. Biometrics for Electronic Eyes: System Authentication with Embedded CMOS Image Sensor. *IEEE Trans. on Consumer Electronics* 62, 3 (2016).
[8] J. Leung et al. 2007. Quantitative Analysis of In-Field Defects in Image Sensor Arrays. In *Proc. DFT.*
[9] L. Lin et al. 2010. Low-Power Sub-Threshold Design of Secure Physical Unclonable Functions. In *Proc. ISLPED.*
[10] M. Liu et al. 2017. A Data Remanence Based Approach to Generate 100% Stable Keys from an SRAM Physical Unclonable Function. In *Proc. ISLPED.*
[11] J. Lukáš et al. 2006. Detecting Digital Image Forgeries using Sensor Pattern Noise. *Proc. SPIE* 6072 (2006).
[12] J. Lukáš et al. 2006. Digital Camera Identification from Sensor Pattern Noise. *IEEE Trans. on Information Forensics and Security* 1, 2 (2006).
[13] Roel Maes et al. 2009. Low-Overhead Implementation of a Soft Decision Helper Data Algorithm for SRAM PUFs. In *Proc. CHES.*
[14] A. Maiti et al. 2010. A Large Scale Characterization of RO-PUF. In *Proc. HOST.*
[15] W. C. Porter et al. 2008. Dark current measurements in a CMOS imager. *Proc. SPIE* 6816 (2008).
[16] E. Quiring and M. Kirchner. 2015. Fragile Sensor Fingerprint Camera Identification. In *Proc. WIFS.*
[17] M. T. Rahman et al. 2014. ARO-PUF: An Aging-Resistant Ring Oscillator PUF Design. In *Proc. DATE.*
[18] G. E. Suh and S. Devadas. 2007. Physical Unclonable Functions for Device Authentication and Secret Key Generation. In *Proc. DAC.*
[19] F. Tehranipoor et al. 2017. DRAM-Based Intrinsic Physically Unclonable Functions for System-Level Security and Authentication. *IEEE Trans. on Very Large Scale Integration Systems* 25, 3 (2017).
[20] D. Valsesia et al. 2017. User Authentication via PRNU-Based Physical Unclonable Functions. *IEEE Trans. on Information Forensics and Security* 12, 8 (2017).