A PUF-based data-device hash for tampered image detection and source camera identification

Zheng, Yue; Cao, Yuan; Chang, Chip-Hong

2019

Zheng, Y., Cao, Y., & Chang, C.-H. (2019). A PUF-based data-device hash for tampered image detection and source camera identification. IEEE Transactions on Information Forensics and Security, 15620-634.

https://hdl.handle.net/10356/137094

https://doi.org/10.1109/TIFS.2019.2926777

A PUF-based Data-Device Hash for Tampered Image Detection and Source Camera Identification

Yue Zheng, *Student Member, IEEE,* Yuan Cao and Chip-Hong Chang, *Fellow, IEEE*

***Abstract*—With the increasing prevalent of digital devices and their abuse for digital content creation, forgeries of digital images and video footage are more rampant than ever. Digital forensics is challenged into seeking advanced technologies for forgery content detection and acquisition device identification. Unfortunately, existing solutions that address image tampering problems fail to identify the device that produces the images or footage while techniques that can identify the camera is incapable of locating the tampered content of its captured images. In this paper, a new perceptual data-device hash is proposed to locate maliciously tampered image regions and identify the source camera of the received image data as a non-repudiable attestation in digital forensics. The presented image may have been either tampered or gone through benign content preserving geometric transforms or image processing operations. The proposed image hash is generated by projecting the invariant image features into a physical unclonable function (PUF)-defined Bernoulli random space. The tamper-resistant random PUF response is unique for each camera and can only be generated upon triggered by a challenge, which is provided by the image acquisition timestamp. The proposed hash is evaluated on the modified CASIA database and CMOS image sensor based PUF simulated using 180nm TSMC technology. It achieves a high tamper detection rate of 95.42% with the regions of tampered content successfully located, a good authentication performance of above 98.5% against standard content-preserving manipulations, and 96.25% and 90.42%, respectively for the more challenging geometric transformations of rotation ($0 \sim 360^\circ$) and scaling (scale factor in each dimension: 0.5). It is demonstrated to be able to identify the source camera with 100% accuracy and is secure against attacks on PUF.**

***Index Terms*—Camera Identification, Digital Image Forensics, Perceptual Image Hash, Physical Unclonable Function.**

I. Introduction

THanks to the advent of information technology, digital images and videos have been increasingly exposed as important information or art carriers in our daily life. Despite easy and cheap to acquire, distribute and store, the threats of abuse are high, which if not carefully solved, will lead to great loss of property, fame, and even life. Images or videos can be cloned illegally, imperceptibly modified using image processing tools or even fabricated with the help of artificial intelligence (AI) to distort the truth to mislead people or clinch wrongful convictions in a court of law. In late 2017, a software called "deep fakes" was anonymously released that uses deep learning to swap the face of a person to create a very realistic fake picture or video. The non-consensual use of this tool to insert celebrity faces onto pornographic videos caused the popular online forum Reddit to shut down its /r/deepfakes subreddit discussion board [1]. This incident raises a red flag, given the prevalent use of surveillance footage to aid criminal investigation and civil litigation. As fraudsters are more adept at using AI, it is imperative to enhance digital (visual) evidences with technologies that can not only detect forgeries (image tampering detection) but also identify the digital device that captures the evidence (source camera identification) to combat anti-forensics.

For image tampering detection problem, the solutions are mainly provided by three types of schemes in the literature [2]: image watermarking [3], [4], digital image forensics [5]–[8] and perceptual image hashing [9]–[11]. The image watermarking-based schemes can detect the distortion based on the assumption that the imperceptibly embedded watermark will also be distorted. However, such methods have fundamental trade-off between perceptual quality degradation and watermark capacity, which limits their sensitivity and robustness against different optimized attacks with a constrained attack distortion [12]. Digital image forensic based schemes aim at blind investigation of malicious tamper with no side information (e.g., watermark or hash values) provided from the original images. The method can be broadly categorized as being visual and statistical. The former is mainly based on visual clues such as inconsistencies in an image while the latter focuses on analyzing the pixel values of the image [13]. However, lacking original data information makes these methods computationally intensive and very time consuming, often with low accuracy of detection. Among all, perceptual image hashing is most effective in tamper detection as it is very sensitive to content-specific modifications but is otherwise robust against normal content-preserving processing like noise, filtering, rotation or scaling. Since such methods depend on a shared secret key for authentication, the security of the whole system will collapse if the secret key is compromised, lost or stolen. It has been demonstrated that storing the secret key in a non-volatile memory (NVM) is vulnerable to data remanence and reverse engineering attacks [14]–[16]. Once the key is cracked, the attacker can easily create a valid hash value for a tampered image.

Source camera identification is mainly achieved using machine learning based methods, which basically follow three steps: image feature extraction, classifier training and image

Manuscript received on January 15, 2019, revised on March 26, 2019 and accepted on June 28, 2019. This project was supported by the Singapore Ministry of Education AcRF Tier 2 grant No. MOE2015-T2-013.

Y. Zheng and C.H. Chang are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798. (Emails: yzheng015@e.ntu.edu.sg, echchang@ntu.edu.sg). Y. Cao is with the College of Internet of Things Engineering, Hohai University, Changzhou 213022, China. (Email: caoyuan0908@gmail.com) (corresponding author: C.H. Chang and Y. Cao.)

source class prediction. It is important to select appropriate features that represent the unique characteristics of the underlying devices. By analysing the structure and processing stages of digital camera, such features can be algorithmically extracted based on the knowledge of lens aberration, sensor imperfection, color filter array interpolation and statistical image features [17]. Most existing works focus on imaging device brand identification [18], [19], which can achieve very high accuracy but fail to distinguish individual devices from the same model and the same brand. Identifying individual camera devices have been increasingly studied in recent years based on photo response non-uniformity (PRNU) pattern [20]–[24], but strict conditions in the acquisition process, the number and content of training images as well as the geometrical synchronization of testing images have to be met in order to achieve high reliability and accuracy. Since those methods require features to be extracted from the images instead of the device, the device source detection accuracy is strongly constrained by the image fidelity and quality. As the image processing methods are transparent, the same approach can also be used by a malicious user to extract the device features from publicly available images. A newer and anti-forensic resistant way of camera device identification uses physical unclonable function (PUF) to directly extract the "device signature" from the pixel array [25], [26]. The authors proposed low-cost CMOS image sensor [26] and dynamic vision sensor [25] based PUFs, which were designed based on the fact that modern integrated circuit manufacturing process will introduce inevitable random variations into individual active pixel elements of identically designed image sensors. Both works have been demonstrated to be able to differentiate every single camera device regardless of model or brand with high accuracy.

Though being widely researched in both directions independently, to our best knowledge, very few works can effectively detect image tampering while at the same time identifying source cameras. This attribute is of particular significance in digital forensics. This concept was proposed in our preliminary works [27], [28], but there are deficiencies in these works. First of all, they are not resilient to geometric transformations like rotation and scaling, which are common content-preserving manipulations in image processing. The experimental results of [27] were obtained from a very small and simple database, which is inadequate to demonstrate its robustness. Moreover, the method proposed in [27] has little noise tolerance on the input challenge due to its avalanche effect on PUF response errors. The extracted features for the authentic regions from the received image have to be exactly matched with the enrolled features, which is very difficult to fulfill in practice. Furthermore, tampered region identification is not considered at all. In [28], physical layer watermarking is used to hide the PUF-based data-dependent hash tag. That work focuses mainly on the robustness of recovering the hidden hash tag transmitted over additive white Gaussian noise channel instead of any content-preserving image processing operations applied directly on the image.

In this work, tamper detection and source camera identification are achieved simultaneously by using perceptual image hashing and PUF in a simple but effective way. The main contributions can be summarized as follows:

1. The proposed perceptual data-device hash is able to imprint an indelible birthmark of the camera for forgery detection of its captured images. A CMOS image sensor based PUF is utilized to generate a device-specific Bernoulli random matrix for the projection of rotation-/scaling- invariant image features to obtain the perceptual hash.
2. The proposed hash is time-, data- and device-dependent, which greatly enhances the system security compared to the existing perceptual hashing methods that are only dependent on the data. The unique and innate device characteristics is directly extracted from the hardware, which greatly simplifies, speeds up and increases the accuracy of individual source camera identification compared to costly, slower and less accurate traditional machine learning based methods.
3. The proposed work solves the secure "key" storage and transmission issues in existing perceptual image hashing scheme for image forensics. Key leakage and hash forgery are prevented as the proposed perceptual image hash is "keyless". Attestation is non-repudiable as the perceptual image hash can only be generated by the timestamp of the image captured through the camera's tamper-resistant image sensor PUF. The threat of server spoofing attacks is eliminated as attestation of tagged image and its origin is performed directly with the acquisition device without the need to store the challenge-response pairs (CRPs) of PUF in trusted server database.
4. An optimal selection of hash dimension and an adaptive threshold is proposed for effective tampered region detection. These improvements maximally discriminate the malicious tampering from content-preserving operations, leading to excellent tamper detection rate and accurate identification of the tampered regions on the tampered images.

The rest of the paper is organized as follows. Some background information on robust image features and PUFs are provided in Section II. The proposed perceptual data-device hash, its generation and how it is used to achieve image tamper detection and source camera identification are detailed in Section III. Section IV presents the experimental setting and parameter optimization. The system performance and results are analyzed in Section V. Section VI concludes the paper.

II. Preliminaries

To keep the paper self-contained, this section briefly introduces the basis of some image and device feature extraction techniques we used for the computation of the proposed perceptual data-device hash.

A. Speeded-up robust features

The speeded-up robust features (SURF) is a robust local feature detector built upon the insight gained from the scale-invariant feature transform (SIFT) descriptor. SURF feature is highly invariant to scale, translation, lighting, contrast and rotation [29], and outperforms SIFT and other popular feature extractors in speed, accuracy, and robustness against different image transformations.

SURF adopts a detector-descriptor scheme, which relies on integral images [30] for fast computation. The detection of scale and orientation invariant interest points is based on the determinant of Hessian (DoH) matrix. Box filter is utilized to approximate the Gaussian second order derivatives, with which the approximated DoH can be calculated as [29]:

$$det(H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \tag{1}$$

where D_{xx} approximates the convolution of the Gaussian second order derivative with an image I at point $c = (x, y)$. Similarly for D_{yy} and D_{xy}. The pre-computed integral image accelerates the computation of D_{xx}, D_{yy} and D_{xy} by using only three additions and four memory accesses, independent of the box filter size. For simplicity, a constant relative weight of 0.9 is used for the filter response to provide the necessary energy conservation between the Gaussian and approximated Gaussian kernels [29].

Such blob responses are calculated at each point of image I over different scales by convolving the same input image with larger filters to obtain a series of filter response maps. The local maxima of the scale-normalized DoH across $3 \times 3 \times 3$ neighborhood with different octaves is found and interpolated both in scale and image space to compensate for the construction error. After which, a predefined threshold is applied to select the strongest feature points from this set of local maxima [31].

The SURF descriptor summarizes pixel information within a local neighborhood. First, the orientation for each feature point is determined by convolving pixels in its neighborhood with the Haar wavelet filter. A square neighborhood centered around the interest point and along the detected orientation is then divided into 4×4 sub-regions. The sum of values ($\sum d_i$) and of magnitudes ($\sum |d_i|$) for both wavelet responses d_x and d_y in the horizontal and vertical directions, respectively of each sub-region are computed as the feature vector entries. By concatenating the 4D feature vectors v_k of all the sub-regions, the i^{th} interest point can be described as a 64-dimension descriptor vector:

$$D_i = [v_1, v_2, \cdots, v_k, \cdots, v_{16}] \in \mathbb{R}^{64} \tag{2}$$

where

$$v_k = \left[\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|\right], k = 1, 2, \cdots, 16 \tag{3}$$

For an image I with m detected interest points, the SURF feature representation can be denoted as

$$F = \{D_{1,s_1}, \cdots, D_{i,s_i}, \cdots, D_{m,s_m}\} \tag{4}$$

$$s_i = [\sigma, s\mathcal{L}, O, L, |det(H)|] \tag{5}$$

where s_i of the i^{th} feature point contains the scale σ, sign of Laplacian $s\mathcal{L}$, orientation O, location L and the DoH magnitude $|det(H)|$.

Finally, SURF exploits a nearest neighbor strategy to perform the image feature matching [32] based on the computed descriptors. MSAC-based technique (described in Sec. II-B) is used to check the geometric consistency.

B. M-estimator Sample Consensus

M-estimator SAmple Consensus (MSAC) algorithm is an improved variant of the random sample consensus (RANSAC) algorithm for effective transformation estimation. RANSAC belongs to the framework of iterative hypothesize-and-verify algorithms, which can be briefly described by the following procedure [33].

First, a minimal sample set (MSS) containing minimal sufficient data items for model parameter determination is randomly selected from the input database. A model is then hypothesized and the model parameters (the transformation) are calculated based solely on the elements from MSS. Next, a consensus set (CS) of inliers is found for this hypothesized model by verifying which elements from the entire database are consistent with the previously estimated model parameters. This hypothesize-and-verify procedure is iterated until the probability of finding a better model falls below a predefined threshold. The best transformation is then estimated by choosing the one with the largest CS ranked according to its cardinality.

$$\text{rank}(CS) \stackrel{def}{=} |CS| \tag{6}$$

Let $\mathbf{D} = \{D_1, D_2, \cdots, D_N\}$ denotes a dataset of input data, θ the estimated transformation and M the model space. RANSAC identifies inliers and evaluates the quality of CS by minimizing the loss function

$$C_M(\mathbf{D}, \theta) = \sum_{i=1}^{N} \rho(D_i, M(\theta)) \tag{7}$$

Each data point D_i is assigned a weight of zero or one by comparing their error functions against a noise threshold δ:

$$\rho(D_i, M(\theta)) = \begin{cases} 0, & |e_M(D_i, \theta)| \leqslant \delta \\ 1, & otherwise \end{cases} \tag{8}$$

The error function $e_M(D_i, \theta)$ is defined as the distance from D_i to $M(\theta)$, i.e.,

$$e_M(D_i, \theta) \stackrel{def}{=} \min_{D_i' \in M(\theta)} \text{dist}(D_i, D_i') \tag{9}$$

where $\text{dist}(a, b)$ is an appropriate distance function between two points, a and b.

To reduce the sensitivity of estimated model parameters to the choice of noise threshold, MSAC modifies $\rho(D_i, M(\theta))$ in Eq. (8) to

$$\rho(D_i, M(\theta)) = \begin{cases} e_M(D_i, \theta), & |e_M(D_i, \theta)| \leqslant \delta \\ \delta, & otherwise \end{cases} \tag{10}$$

MSAC improves the efficiency of RANSAC by redescending. It scores the inliers according to their fitness to the model and assigns the outliers a constant weight. The number of iterations τ_{stop} [33] can be set to:

$$\tau_{stop} = \left\lceil \frac{\log \varepsilon}{\log(1 - q)} \right\rceil \tag{11}$$

where q represents the probability of sampling a MSS from $\mathbf{D}$ that produces an accurate estimation of the model parameters, and ε is a predefined probability threshold (a.k.a alarm rate) such that the probability of no unbiased MSS is picked after τ_{stop} iterations is at most ε.

C. Physical Unclonable Function

Physical unclonable function (PUF) [34] is a unique hardware-oriented security primitive that does not rely on key-based algorithmic intractability or hard-to-solved mathematical problems as the basis for trust establishment. PUF harnesses the subtle mismatches or disorder in electrical properties of identically designed circuits from inevitable and uncontrollable physical parameter variations in nano-scale device manufacturing process. A PUF can be mathematically modelled as an irreversible mapping of an input challenge to an output response. The challenge-response pair (CRP) is unique for different dies of the same wafer and across wafers, making PUF an ideal "device fingerprint". Besides, the response of the PUF can only be generated upon request by an input challenge, which avoids the need to hardcode the device identity or store the secret key locally on NVM. Because of this, PUF possesses tamper-aware or tamper-evident property as any modification to the PUF circuit will change the original CRP mapping and render a genuine device unable to be authenticated.

In this work, we make use of PUF to generate device-specific random space for the projection of image features. Strong PUF [34], such as arbiter PUF, has an exponential number of challenges relative to its number of bit-slices. While the practically inexhaustible number of challenges is good to ensure freshness of CRPs against replay and man-in-the-middle attacks in device authentication, the responses to different challenges are not mutually independent as they are generated based on the linear additive path delay of cascaded bit-slices. As the number of CRPs are significantly larger than the unknown device parameters that contribute to the one-way function, strong PUFs are also potentially vulnerable to cloning attack by machine learning. Memory-based PUF is a typical weak PUF of limited number of challenges. As each response bit is independently generated by a memory cell, its number of challenges scales linearly instead of exponentially with the number of addressable bit-cells. Such PUF that can be intrinsically reused as another functional module in computer system is particularly desirable to avoid the overhead of a dedicated chip area reserved for chip-unique random response generation. The CMOS image sensor used for digital imaging has a similar array structure of independently accessible pixel elements. Modern CMOS image sensors have great resolution. By reusing the integrated CMOS image sensor of an image acquisition device for PUF response generation, the number of pixels is more than sufficient to provide the required random mapping space.

III. Proposed PUF-based Data-device Hash

To detect maliciously tampered, unscrupulously manipulated and fabricated images without restricting benign image processing and analysis, we proposed to tag the image with a distinguished provenance to irreversibly and non-repudiable bind the information integrity, source authenticity and acquisition timestamp. The tag can be generated by integrating the sensor-level device information with the perceptual invariant image features at the time of capture. The extraction and unification of the image features and "device fingerprint" are detailed in this section.

A. Robust Feature Extraction

Salient features extracted from the captured image for the generation of the proposed perceptual data-device hash should satisfy the following requirements. First, distinguishable image features and acquisition device characteristics should be reliably and independently extracted before their fusion. To reduce the computational burden and improve the accuracy, unique device features are to be extracted directly at sensor-level instead of by statistical processing or learning from a large pool of images captured by the device. Secondly, the extracted image features should be invariant to common image processing operations like rotation, filtering and gamma correction, and have good tolerance against inevitable noise contamination during data processing or transmission. Last but not least, as tampering tends to focus on dominant local instead of global features, the tampered regions should be identifiable from the change in dominant block features. To fulfill these requirements, we extract the image features from rotation-invariant SURF and adjoint block-based DCT concatenated features, and the "device fingerprint" from the CMOS image sensor based PUF.

1) Rotation-Invariant SURF Features: In order to achieve efficient transmission, only a small constant number of the strongest SURF features are kept. For images that have feature points more than a predetermined threshold T_f ($T_f = 100$ in our experiment), it will be truncated directly from the $(T_f+1)^{th}$ feature in descending order of salience. Assuming t effective feature points are detected in an $M \times N$ source image, the 64-D SURF features are denoted as:

$$F = \{D_{1,s_1}, \cdots, D_{i,s_i}, \cdots, D_{t,s_t}\} \in \mathbb{R}^{64\times t}, t \leq T_f \quad (12)$$

where s_i is the descriptive information for each SURF feature point as denoted in Eq. (5).

2) Adjoint Block-based DCT Features: Discrete cosine transform (DCT) (typically DCT-II) is a popular block-based feature extraction method with strong "energy compaction" property. It is not only robust against cropping, noising, filtering and sharpening, but also has good computational efficiency [35]. Adjoint block-based DCT concatenated feature is proposed in [28] to increase the energy concentration on local features. It can be easily obtained by concatenating the DCT-II coefficients of small neighboring sub-blocks. An $M \times N$ image is first divided into non-overlapping 8×8 elementary blocks ($eblocks$) before applying the DCT-II transform on each $eblock$. Since most of the signal information tend to be concentrated on a few low-frequency components, only the first 50% of the DCT coefficients are kept and zigzagged to obtain the $eblock$ feature vector $f \in \mathbb{R}^{32}$. The $cblock$ is then formed by combining four neighboring $eblocks$, whose feature vectors are concatenated together to form the $cblock$ feature $F \in \mathbb{R}^{128}$ without compromising the resolution and energy of localized features.

$$\begin{aligned} F_{i,j} = [f_{2i-1,2j-1}, f_{2i-1,2j}, f_{2i,2j-1}, f_{2i,2j}] \in \mathbb{R}^{128} \\ i \in \{1, 2, \cdots M/2\}, j \in \{1, 2, \cdots N/2\} \end{aligned} \quad (13)$$

where (i, j) is the row and column indexes of the $cblock$. $f_{2i-1,2j-1}$ denotes the $eblock$ feature vector extracted from

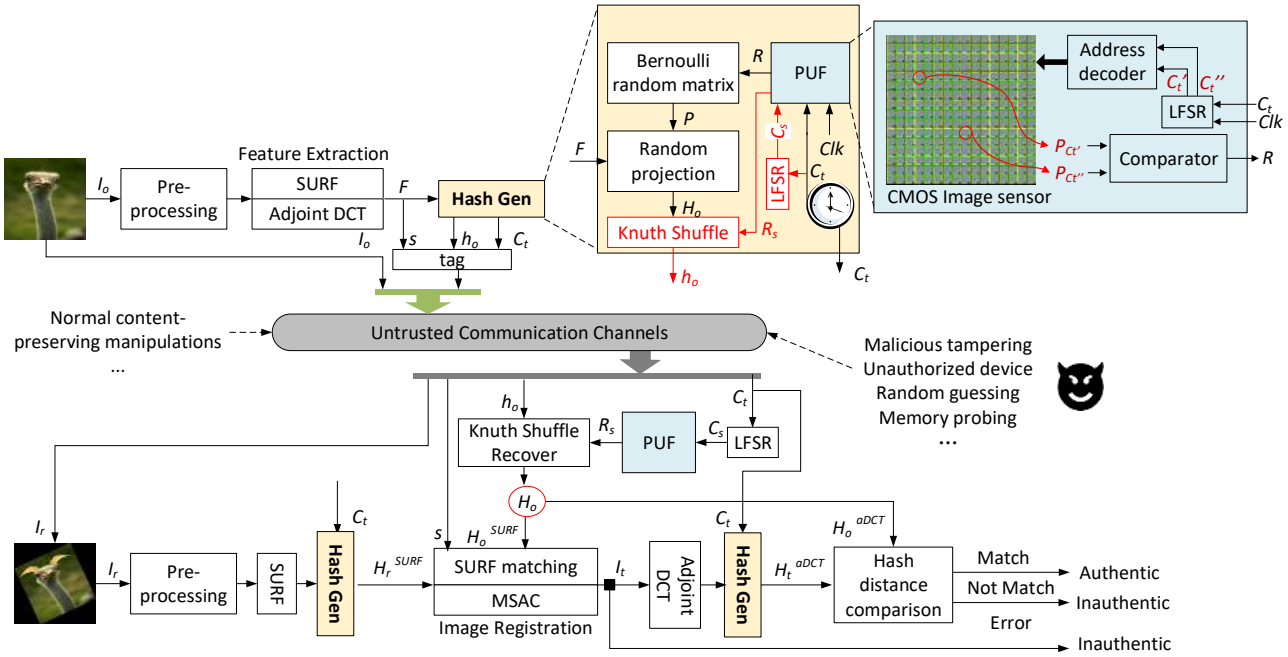

Fig. 1. Framework of the proposed perceptual data-device hash. Noted that the "Hash Gen" module in the verification phase includes everything except the shuffle path, which is printed in red, in the "Hash Gen" block.

the $eblock$ that resides in the $(2i-1)^{th}$ row and $(2j-1)^{th}$ column of the original image I.

3) CMOS Image Sensor Based PUF: To avoid sophisticated statistical image processing techniques, CMOS image sensor based PUF [26] uses the innate fixed pattern noise (FPN) of active pixel sensor array for camera or imaging device identification. FPN as a whole refers to the random variations of output pixel voltage values of an image sensor under uniform illumination or even in complete darkness. This phenomenon is induced by the small deviations in individual pixel responsivity of the sensor array contributed by the transistor size and interconnect mismatches as a consequence of random lithography variations. In this work, the timestamp C_t generated at the time of image capture is converted into a digital word and applied as a challenge to the PUF. As shown in Fig. 1, C_t is applied as a seed of a linear feedback shift register (LFSR) to generate the internal challenges C_t' and C_t'' at the control of a clock Clk. C_t' and C_t'' are applied to the CMOS image sensor array to locate a pair of active pixel sensors. Their corresponding reset voltages $P_{C_t'}$ and $P_{C_t''}$ are read out by disabling the correlation double sampling circuit and then compared to generate a response bit R_i. Unreliable response bits with absolute reset voltage difference less than a given threshold are discarded. An L_R-bit response R can be obtained by clocking the LFSR at least $2L_CL_R$ cycles, where L_C is the bit length of challenge. This CRP mapping is unique to each PUF instance, and its high uniqueness ensures that individual camera can be distinguished with high accuracy regardless of model type or brand.

B. Perceptual Data-Device Hash Generation

This section elucidates how the extracted image features and "device fingerprint" are indivisibly fused into a compact perceptual data-device hash. Several design objectives are to be met. First, to prevent the key leakage problem, which is a major weakness of conventional perceptual image hashing, the hash should not rely on a persistently stored local secret key for its generation. Secondly, the integration of both data and device information should provide an acceptably strong discriminative power for tamper detection and source camera identification. Last but not least, the hash should be sufficiently compact and can be computed efficiently.

To fulfill the above objectives, the proposed data-device hash is generated by projecting the image features into a device-unique random space. The latter is defined by the response of the CMOS image sensor PUF, which can only be generated when the PUF is stimulated by a timestamped challenge C_t. Random projection (RP) is a widely used efficient dimension-reduction technique. The key idea stems from *Johnson-Lindenstrauss (JL) Lemma* [36], which can be stipulated as: Given $\epsilon \in (0,1)$, if $m \geq O(\epsilon^{-2}\log Q)$, then every high-dimensional dataset $X \in \mathbb{R}^n$ of Q points can find its Lipschitz mapping $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ such that

$$(1-\epsilon) \parallel u-v \parallel^2 \leq \parallel f(u)-f(v) \parallel^2 \leq (1+\epsilon) \parallel u-v \parallel^2 \quad (14)$$

for any $u, v \in X$.

Three commonly used matrices Φ that have been proven to be qualified for the implementation of f in Eq. (14) are listed below [37]:

1. iid samples from $\mathcal{N}(0, 1/m)$;

2. independent realizations of ± 1 Bernoulli random variables:

$$\Phi_{i,j} = \begin{cases} \frac{+1}{\sqrt{m}}, & \text{probability} = 0.5 \\ \frac{-1}{\sqrt{m}}, & \text{probability} = 0.5 \end{cases} \tag{15}$$

3. a related distribution that yields the matrix:

$$\Phi_{i,j} = \begin{cases} +\sqrt{\frac{3}{m}}, & \text{probability} = \frac{1}{6} \\ 0, & \text{probability} = \frac{2}{3} \\ -\sqrt{\frac{3}{m}}, & \text{probability} = \frac{1}{6} \end{cases} \tag{16}$$

Since PUF response is ideally a random vector of binary bits with uniform distribution, we proposed to specify the projection space by a PUF response driven Bernoulli random matrix. The entries of Bernoulli random matrix are set to $+1/\sqrt{m}$ or $-1/\sqrt{m}$ when the PUF response bit R_i is 0 or 1, respectively, to produce the device-unique random space P.

$$P_i = \begin{cases} \frac{+1}{\sqrt{m}}, R_i = 0 \\ \frac{-1}{\sqrt{m}}, R_i = 1 \end{cases} \tag{17}$$

To generate an m-dimensional perceptual data-device hash, a PUF response of length $n \times m$ is collected from the CMOS image sensor, with n being the dimension of the raw image feature vector. By projecting the SURF features F of Eq. (12) and the adjoint DCT features $F_{i,j}$ of Eq. (13) into this PUF-specified Bernoulli random matrix, a data-device hash H can be generated as follows:

$$\begin{aligned} H &= [H^{SURF}, H^{aDCT}] \\ &= [P_{f1}^{\mathrm{T}} D_{1,s_1}, \cdots, P_{f1}^{\mathrm{T}} D_{i,s_i}, \cdots, P_{f1}^{\mathrm{T}} D_{t,s_t}, \\ &\quad P_{f2}^{\mathrm{T}} F_{1,1}, \cdots, P_{f2}^{\mathrm{T}} F_{M/2,N/2}], \quad t \leq T_f \end{aligned} \tag{18}$$

where the superscript T is a matrix transpose operator. P_{f1} and P_{f2} correspond to the random matrix P generated for SURF and adjoint DCT features, respectively, which may have different feature dimensions.

Tagging the image with the calculated data-device hash H directly is vulnerable to lunchtime attack. Due to the linearity of random projection, a malicious user can invert the projection by carefully crafting the image features into a full rank matrix with the hashes collected from temporary possession of the device. Once the random projection matrices, P_{f1} and P_{f2}, are recovered, they can be used to generate valid hashes for other images even without the correct device. To solve this problem, the hash H is randomly shuffled by a modified Knuth Shuffle algorithm, which can be realized using the PUF response as the seed to an unbiased random integer generator (e.g., LFSR or irand() function in C program), as shown in Algorithm 1. To ensure that its output is unbiased, one bit of the seed is flipped in each round of the for loop to initiate a new random cycle. Additionally, the seed R_s can be generated by applying a new challenge derived from C_t. The bit length of R_s has to be sufficiently long, e.g., at least 128 bits, to prevent brute force attack.

The output of Knuth Shuffle is tagged to the image for transmission and storage. With the correct device held by a legitimate user, the seed R_s can be regenerated by the embedded PUF. With the correct seed, the direct hash H can be recovered for further verification. As the seed is internally generated by the PUF, it is impossible for the adversary to recover a valid H for his/her tagged image to pass the authentication.

Algorithm 1 Knuth Shuffle

Input: PUF response R_s, direct hash H of length l
  for i = l downto 2 **do**
    j = (unbiasedRandIntGen(R_s) mod $(i-1)$) +1;
    swap $H(i)$ and $H(j)$
  end for

C. Image Tampering Detection and Source identification

Fig. 1 shows the framework of the proposed PUF-based perceptual data-device hash in a digital forensic application scenario. As shown in Fig. 1, once an image of interest I_o is captured by a camera, its SURF and adjoint DCT features F will be extracted. At the same time, a timestamp C_t is generated and applied to the embedded CMOS image sensor PUF to obtain a response R, which is further processed to produce the Bernoulli random matrix P. A data-device hash H_o is then calculated by projecting F into the P space. A shuffle challenge C_s derived from C_t is then applied to the PUF to extract a 128-bit R_s as the seed to the Knuth Shuffle module. Finally, the descriptive information s of the SURF feature, the shuffled data-device hash h_o as well as the timestamp C_t are tagged on the original image I_o. The proposed system is able to validate the received image authenticity, locate any small tampering regions and identify the source device based on the received image I_r, descriptive information s, hash H_o and timestamp challenge C_t using the claimed device. The authentication framework comprises an image registration stage and a hash distance comparison stage.

1) Image Registration: The received image may have previously undergone certain geometric deformation like rotation or scaling that causes its coordinates to deviate from the original one. Therefore, SURF feature projected hash H^{SURF} is used for image registration in order to reproduce H^{aDCT} for hash distance comparison.

To perform authentication, the verifier inputs the dubious image I_r to the claimed device. The tagged hash h_o, the descriptive information s and the corresponding challenge C_t are extracted from I_r. Firstly, the extracted challenge C_t will be fed into a LFSR to generate a new shuffle challenge C_s, which is applied to the embedded PUF to obtain the 128-bit Knuth Shuffle seed R_s. With R_s, the verifier is able to recover the unshuffled hash H_o(=[H_o^{SURF},H_o^{aDCT}]) from h_o. Using the same pre-processing as the original image I_o, the SURF features F_r of I_r are also extracted. Meantime, the challenge C_t stimulated PUF response R (divided into R_{f1} and R_{f2}) is applied for Bernoulli random space P (P_{f1} and P_{f2}, respectively) generation. By calculating $P_{f1}^{\mathrm{T}} F_r$, H_r^{SURF} is regenerated. Taking H_r^{SURF} and the original hash tag H_o^{SURF} as inputs, matched points between the original image and the received image are found. As it is also possible

that the received image has not undergone any geometric transformation, to prevent the loss of precision, the presence of geometric transformation is first ascertained by comparing the corresponding locations of the matched points of I_o and I_r recovered from H_o^{SURF} and H_r^{SURF}, respectively. They should be identical ideally if there is no geometric transformation between I_o and I_r. However, standard image processing operations like denoising or filtering and SURF matching algorithm accuracy can change the locations of those detected matched points slightly. In the proposed method, I_r is said to be in the same coordinate as I_o if the matched point pairs with absolute location deviation of less than 5 occupy more than 50% of the total matched pairs, i.e.,

$$\begin{aligned} &L_{I_o} = matchedPtsOriginal.Location; \\ &L_{I_r} = matchedPtsReceived.Location; \\ &\frac{\#(abs(L_{I_o} - L_{I_r}) \leq 5)}{\#(All\_matchedPtsPairs)} \geq 0.5 \end{aligned} \tag{19}$$

where $\#(A)$ is the cardinality of dataset A. In this case, I_t in Eq. 20 will be directly assigned I_r.

Once geometric transformation is detected in the received image I_r, MSAC is performed to find the best affine transformation that maps the most matched points from I_r to I_o. As a result, the recovered image I_t can be calculated by:

$$I_t = \theta I_r = \begin{bmatrix} a_1 & a_2 & b_1 \\ a_3 & a_4 & b_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r \\ y_r \\ 1 \end{bmatrix} \tag{20}$$

where θ denotes the general description of the returned transformation matrix, in which $[b_1\ b_2]^{\mathrm{T}}$ represents the translation vector and the parameters $a_i (i = 1, 2, 3, 4)$ defines the transformations like image rotation and scaling.

Fig. 2. An example of wrong matching: (a) original image I_o; (b) received image I_r; (c) recovered image I_t.

There is one exceptional case of image registration failure due to overtly-tampered, completely different images or the use of a wrong device (see more details in Sec. V-C). This situation may lead to limited or even no matched SURF points found. Under this circumstance, even if the SURF feature detection and matching are conducted as normal, the recovered images are probably distorted as shown in Fig. 2. This abnormality in feature matching can be detected by the number of absolute black pixels. If the number of absolute black pixels in I_t exceeds certain threshold (10% in our experiments) of the total image pixels, it is deemed as an image registration failure and the received image will be rejected immediately.

2) Hash Distance Comparison: Other cases of maliciously tampered images that pass image registration (usually small tampering) can be detected in the hash distance comparison phase. If the image registration is successful, the recovered image I_t that has the same coordinate system as I_o will be obtained. By applying adjoint DCT feature extraction and PUF-based random projection (projection space: P_{f2}), the hashed adjoint DCT features H_t^{aDCT} are obtained. The Euclidean distance e between H_t^{aDCT} and H_o^{aDCT} is calculated over each $cblcok$ and compared with a tamper threshold τ_e to detect the tampering:

$$\begin{aligned} e_{i,j} &= \sqrt{((H_t)_{i,j}^{aDCT} - (H_o)_{i,j}^{aDCT})^2} \\ i &\in \{1, 2, \cdots M/2\}, j \in \{1, 2, \cdots N/2\} \end{aligned} \tag{21}$$

The outcomes of these two stages are shown in Fig. 3.

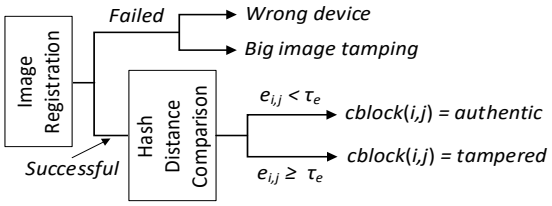

Fig. 3. Received image authentication of proposed data-device hashing.

Adaptive threshold: An adaptive threshold is proposed to increase the robustness of tampered image detection. The problem with using a fixed threshold for all the images in conventional perceptual image hashing [10], [28] is the distribution of distance e varies among different original-received image pairs. To achieve better tamper detection performance, the threshold τ_e is made adaptive to the image pair. Taking into consideration that the Euclidean distance between the received and original images is larger for the tampered $cblocks$ than the untampered $cblocks$, to extract the authentic $cblock$ information, the median Euclidean distance of the image pair is calculated and denoted as $\tilde{e}(I)$. For simplicity and ease of computation, the adaptive tamper threshold $\tau_e(I)$ is determined by a linearly separable hyperplane by mapping the median distance $\tilde{e}(I)$ and the maximum Euclidean distance e_{max} across all $cblocks$ of an image pair I to the same space for different original-content preserving and original-tampered image pairs. Thus,

$$\tau_e(I) = a \times \tilde{e}(I) + b \tag{22}$$

where a and b are the coefficients that can be empirically determined. The linearity assumption of separable hyperplane was found to have no significant negative impact from the results of our experiments. The tampered regions of I can be more conspicuously identified by the $cblocks$ with this image pair dependent adaptive threshold $\tau_e(I)$ than a fixed threshold.

IV. Experiment Setting and Key Parameter Optimization

A. Dataset Preparation

Modified CASIA database is used in this work for performance testing. The ground truth images are taken from the

CASIA image tampering detection evaluation (ITDE) v1.0 database [38], which contains images from eight categories (animal, architecture, article, character, nature, plant, scene and texture) of size 384×256 or 256×384. Instead of directly using the tampered image set from CASIA ITDE v1.0, the tampered versions of those authentic images are selected from CASIA ITDE V2.0, which are more challenging and comprehensive since it considers post-processing like blurring or filtering over the tampered regions to make the tampered images appear realistic to human eyes. For one authentic image, there may be several tampered versions in the CASIA ITDE v2.0 dataset. To increase the diversity, only one tampered version is kept for each authentic image. As a result, the modified CASIA database contains 400 (8 categories $\times$ 50 per category) authentic images and their corresponding tampered versions.

According to CASIA ITDE v2.0, the tampered images are generated using crop-and-paste operation under Adobe Photoshop on the authentic images, and the tampered regions may have random shapes and different sizes, rotations or distortions. In order to evaluate the proposed system performance over content-preserving manipulations, we enrich the modified CASIA dataset by adding content-preserving manipulations to the authentic images using Matlab and ImageJ. Common image processing techniques like rotation, scaling, filtering and JPEG compression, and unavoidable process/transmission noises like Gaussian, Salt&Pepper and speckle noise are considered. Furthermore, the abovementioned content-preserving manipulations are also applied to the tampered dataset to evaluate if their combination can evade detection. As a result, the modified CASIA database D contains:

1) D_{au}: 400 authentic images in 8 categories, each with 50 images;
2) $D_{tampered}$: 400 tampered images corresponding to the authentic ones from D_{au};
3) D_{au_cp}: 3600 (400×9) images generated by adding a single content-preserving manipulation (9 types: Gaussian noise, salt&pepper noise, speckle noise, Gaussian filter, motion blur, JPEG compression, gamma correction, rotation and scaling) to every image of D_{au};
4) $D_{tampered\_cp}$: 3600 (400×9) tampered images by applying those 9 content-preserving manipulations listed in 3) to the images of $D_{tampered}$.

Fig. 4 shows D_{au}, $D_{tampered}$ and D_{au_cp} with their corresponding parameters and tools used. Since the experiment setting has to be defined before the system is deployed, 160 authentic images and their corresponding manipulations from $D_{tampered}$, D_{au_cp} and $D_{tampered\_cp}$ are used as training dataset D_{train} in this section to extract the optimal parameters, while the remaining 240 authentic images and their corresponding manipulations are used as the testing dataset D_{test} for performance evaluation in Sec. V.

For the authorized cameras used in our experiment, their CRPs were simulated by eight PUF instances for 128×128 CMOS image sensor array using 180nm TSMC CMOS technology process design kit in Cadence environment. The design of CMOS image sensor PUF of [26] is adopted. The PUF challenge is 16 bits while the response bit length will be determined after the optimal hash dimension has been determined in Sec. IV-B. Monte-Carlo simulated results of PUF designed for 64×64 CMOS image sensor array were first validated by the real silicon data measured from five 64×64 CMOS image sensor array PUF chips fabricated also in the 180nm CMOS technology [26]. More instances of PUF designed for the larger 128×128 CMOS image sensor array were then simulated to evaluate the PUF quality. The fractional Hamming distance distribution from responses generated by 42 simulated PUF instances is Gaussian distributed with mean and standard deviation of 0.5002 and 0.0039, respectively. The 20 blocks of 35k response bits each also passed the NIST randomness tests. The reliability after discarding 5.5% of total pixel pairs by thresholding is $> 98.17\%$ over a temperature range of -45 to $95°C$ and 100% with $\pm11\%$ supply voltage variations.

B. Hash Dimension Selection

As introduced in Sec. III-B, the hash dimension m is determined by the PUF-based Bernoulli random matrix P, which has a size of $n \times m$, with n being the original feature dimension. Let n_1 and n_2 be the dimensions of SURF feature and adjoint-DCT feature, respectively per adjoint block, and p be the projection rate (projection rate refers to the ratio of projected feature dimension to the original feature dimension), the final hash dimension can be calculated by $m = m_1 + m_2 = round(p \times n_1) + round(p \times n_2)$. According to JL Lemma, lower ϵ means better preservation of Eq. (14), which can be ensured by increasing m (or equivalently p). However, increasing m will reduce the hash compactness and require larger PUF size. Since $n_1 = 64$ and $n_2 = 128$, to avoid a large random projection matrix ($n_1 \times m_1$ and $n_2 \times m_2$), the upper bound of the projection rate is set to 0.3 to keep the size of random matrix P below 6080 bits. In order to select an optimal hash dimension, we tested the maximum Euclidean distance e_{max} across $cblocks$ for each image pair in both content-preserving and tampered cases with p of 0.05, 0.1, 0.2 and 0.3.

$$e_{max} = \max_{i\in[1,M/2],j\in[1,N/2]} e_{i,j} \tag{23}$$

To obtain the optimal hash dimension, e_{max} of the original-received image pairs, D_{au}-D_{au_cp} and D_{au}-$D_{tampered}$, in D_{train} are tested. Fig. 5 is the notched boxplot that shows the e_{max} distribution of 9 content-preserving cases and the tampered case, where different colors are used to indicate the different projection ratios of p. Each colored box with a notch around the central mark represents the interquartile range (IQR). The notch represents the 95% confidence interval for the median (the central mark). If the notches between two random distributions in the boxplot do not overlap, it can be concluded that, with 95% confidence, their true medians differ. For a better separability between the content-preserving cases and the tampered case, p should be optimally selected to ensure that there is enough margin to determine a threshold $\tau_e(I)$ of an original-received image pair to discriminate between different content-preserving cases and the tampered case. If e_{max} of an image pair exceeds $\tau_e(I)$, the received image will be rejected and those $cblocks$ of the received image of I that

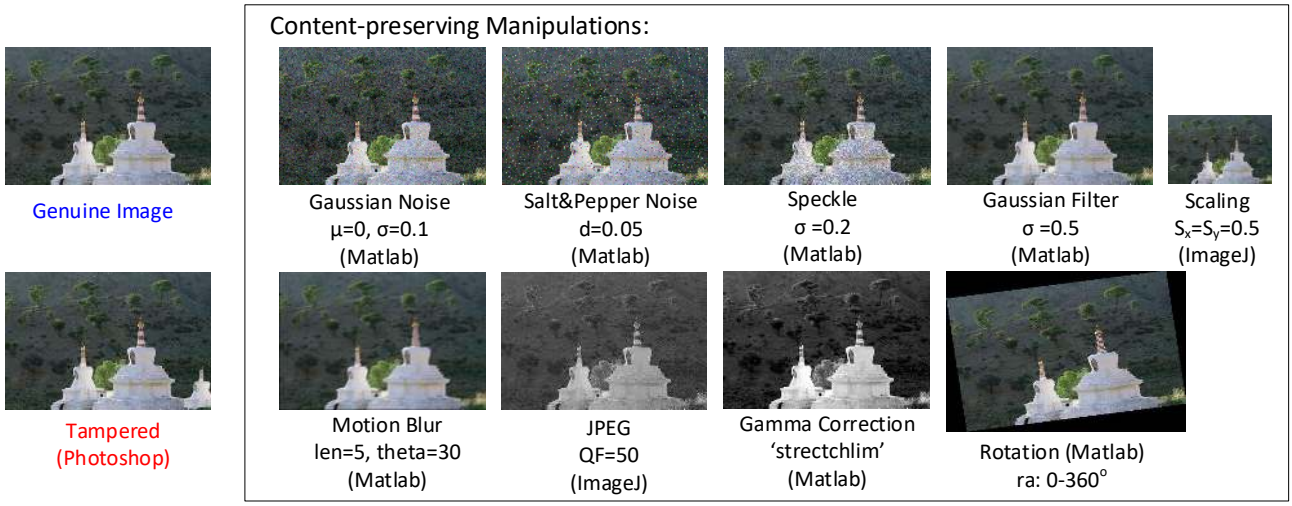

Fig. 4. An example image in the dataset. Genuine image (in blue font) with its 9 content-preserving manipulations (in black font) and 1 tampered version (in red font). The format of the label: manipulation technique, parameters, and the tool used are annotated in brackets. The notations used for the parameters are – μ: mean; σ^2: variance; d: noise density; ra: rotation angle; QF: quality factor; S_x, S_y: scaling factor in x and y dimensions; 'stretchlim': a Matlab function that can automatically achieve the optimal gamma correction.

have $e > \tau_e(I)$ are identified as the tampered regions. The discriminability between each content-preserving case and the tampered case can be observed by comparing the IQRs and medians of the e_{max} distributions for the original-received image pairs in different cases. The experimental results in Fig. 5 show that when $p = 0.05$ ($m_1 = 3$, $m_2 = 6$), geometric transformations including scaling (scale factor in each dimension: 0.5) and rotation (factor: $0 \sim 360°$) lead to large e_{max} in content-preserving cases. The IQRs of their e_{max} distributions even overlap with that of the tampered case, which leave insufficient margin for thresholding. When the hash dimension is too small, there are insufficient image features to substantiate Eq. (14). The e_{max} distributions for $p = 0.1, 0.2$ and 0.3 have comparable IQRs and medians in the tampered case as well as in each of the content-preserving cases. More importantly, for each of these projection ratios, there is sufficient gap between the notches of the tempered distribution and any of the content-preserving distributions. As the device key length will increase proportionally from 960 to 2048, 4160 and 6080, respectively as p increases from 0.05 to 0.1, 0.2 and 0.3, to keep the hash compact, $p = 0.1$ is selected.

C. Adaptive Threshold Setting

Once the optimal hash dimension has been set by $p = 0.1$, the tamper detection threshold τ_e can be determined by finding a linearly separable hyperplane in $\tilde{e}(I)$ and e_{max} of image I for all cases of original-received image pairs mentioned in Sec.IV-B. Fig. 6 shows that the e_{max} values of the tampered case mainly cluster in the top of the inclined plane of those content-preserving cases. It is evident that any horizontal line (i.e., a fixed threshold) is incapable of satisfactorily separating the benign cases from the malicious case. A simple adaptive threshold can be derived from a linearly separable hyperplane to distinguish these two classes by any linear classification method such as Bayesian Linear Classifier (BLC). The green line $y = 3.3 \times \tilde{e}(I) + 1030$ shown in Fig. 6 denotes the threshold found using the BLC-based classification method.

Under rare circumstances where the tampered regions of the received image are exceptionally large or the received image is a completely different image of the original, $\tilde{e}(I)$ can be too large to cause the calculated threshold τ_e to exceed e_{max}, resulting in false acceptance as the Euclidean distance of all $cblocks$ of the malicious image pair will fall below the threshold. It is observed that $\tilde{e}(I)$ of content-preserving cases mainly cluster around the range below 500. This problem can be easily resolved by putting a limit on the adaptive threshold value once the $\tilde{e}(I)$ exceeds 500. As e_{max} corresponds to the worst tampered region in the received image, the separation line y can be moved downwards to detect more tampered $cblocks$. Lowering line y too much can also lead to higher false rejection rate (FRR) of content-preserving cases. A balance is struck by setting an offset boundary (confined by the magenta dash lines y_1 and y_2 in Fig. 6) for the separation line. By keeping the gradient and varying the intercept of line y with a step size of 100, the false acceptance rate (FAR) in the tampered case and the FRR in the content-preserving cases are measured and presented in Fig. 7. The experimental results show that the error rates increase rapidly for the tampered cases but decrease modestly for the geometric transformation (rotation and scaling) with the rise of line intercept. The error rates remain relatively constant for other content-preserving manipulations. To maximally detect all tampered regions with minimal negative impact on the error rates in all cases, the adaptive threshold in Eq. (22) is set to:

$$\tau_e(I) = \begin{cases} 3.3 \times \tilde{e}(I) + 730, & \tilde{e}(I) \leq 500 \\ 2380, & \tilde{e}(I) > 500 \end{cases} \quad (24)$$

V. Performance and Discussion

In this section, the proposed system performances are evaluated using the testing dataset D_{test}.

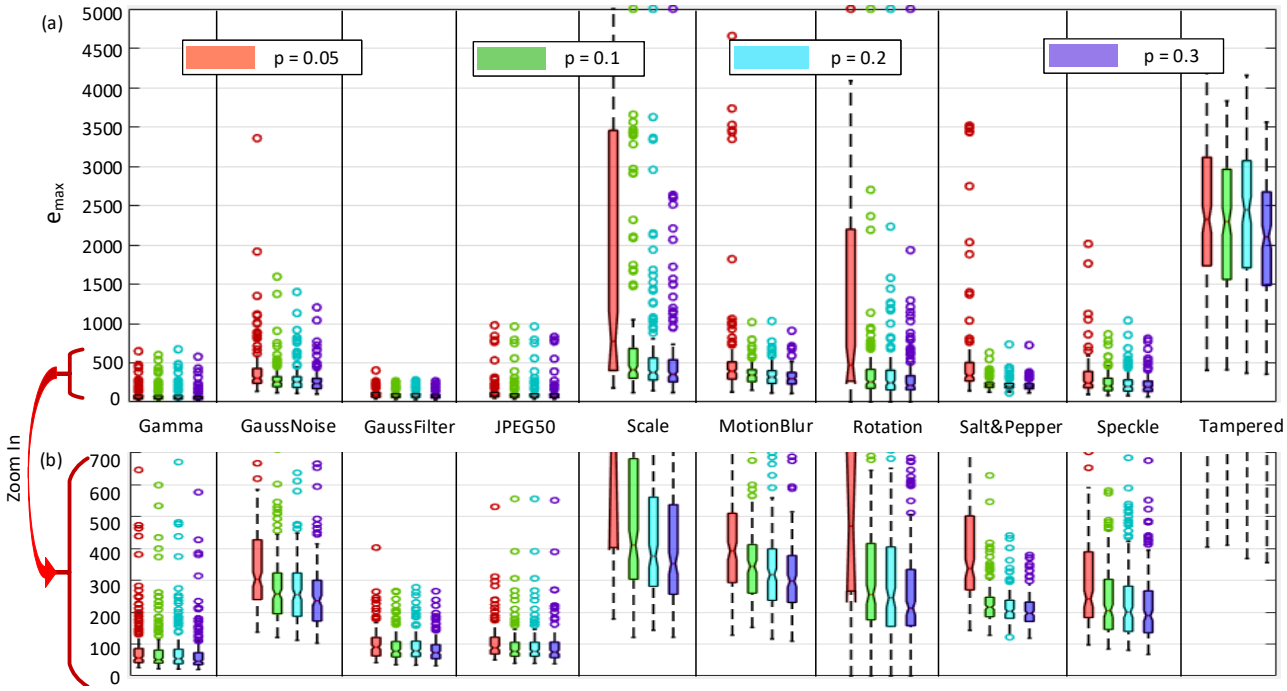

Fig. 5. (a) e_{max} distribution for various cases of manipulations with different projection rates; (b) Enlarged details for the Y axis (e_{max}) range between $0 \sim 700$.

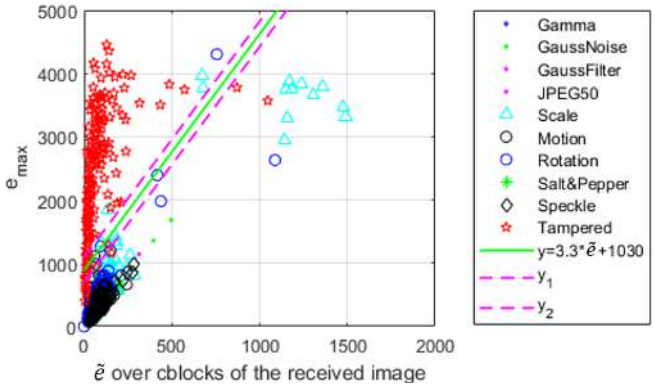

TABLE I
PERCEPTUAL ROBUSTNESS TEST.

Manipulations	Parameters	FRR
Gaussian noise	$\mu = 0, \sigma = 0.1$	0.0042
Salt&Pepper noise	$d = 0.05$	0.0083
Speckle noise	$\sigma = 0.2$	0.0042
Gaussian Filter	$\sigma = 0.5$	0.0000
Motion Blur	$len = 5, \theta = 30$	0.0083
JPEG compression	$QF = 50$	0.0125
Gamma Correction	'stretchlim'	0.0083
Rotation	$ra = 0 \sim 360°$	0.0375
Scaling	$S_x = S_y = 0.5$	0.0958

Fig. 6. Adaptive threshold determination.

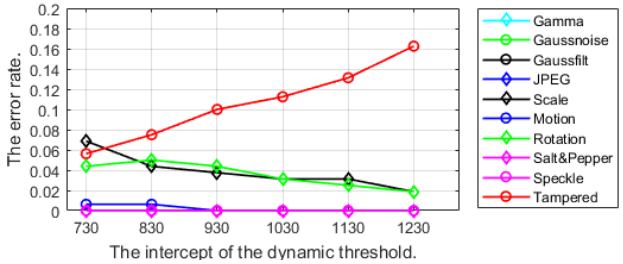

Fig. 7. The error rates (FAR in the tampered case and FRR in the content-preserving cases) with adaptive threshold obtained by varying the intercept of the BLC separation line.

A. Perceptual Robustness

Perceptual robustness tests the tolerance of the perceptual image hash to content-preserving manipulations such as noising, blurring, JPEG compression and so on. The received images that have undergone those content-preserving operations listed in Fig. 4 should be classified as authentic. The FRR of each content-preserving case is measured to evaluate the perceptual robustness. Lower FRR indicates a better perceptual robustness. Table I shows that with $\tau_e(I)$ of Eq. (24) for $p = 0.1$, the proposed method achieves a very low FRR of $< 1.5\%$ for most of the content-preserving cases except rotation and scaling, which have slightly higher FRRs of 3.75% and 9.58%, respectively. For the use case of image forensics, the authentication result can be supplemented by a score, which could be obtained by relating $d_{i,j} = |e_{i,j} - \tau_e|$ to a pre-defined confidence level table, to indicate the confidence of the accept or reject decision. For an authentic or a tampered image with the confidence level score lower than an acceptable threshold, further evidences are needed to support the decision.

B. Tamper Detection and Location

Tamper detection rate (TDR) measures the ability of an image hash in detecting the malicious manipulations of the received image. A good perceptual hash should have not only higher TDR, but also capable of correctly locating the tampered regions. Since the tampered image may also undergo normal image processing, such manipulations should have negligible effect on the TDR. These desirable properties are evaluated for the proposed perceptual image hash.

TABLE II
TAMPER DETECTION TESTS.

Manipulations	Parameters	TDR
Tampered only	$--$	0.9542
+ Gaussian noise	$\mu = 0$, $\sigma = 0.1$	0.9125
+ Salt&Pepper noise	$d = 0.05$	0.9250
+ Speckle noise	$\sigma = 0.2$	0.9125
+ Gaussian Filter	$\sigma = 0.5$	0.9542
+ Motion Blur	$len = 5$, $\theta = 30$	0.9250
+ JPEG compression	$QF = 50$	0.9500
+ Gamma Correction	'stretchlim'	0.9542
+ Rotation	$ra = 0 \sim 360°$	0.9250
+ Scaling	$S_x = S_y = 0.5$	0.9042

Table II shows a high TDR of 95.42% over 240 test image pairs. With content-preserving manipulations on tampered image, the TDR is still $\geqslant 90\%$ though slightly lower than the "tampered only" case. The minor reduction in TDR is ascribed to the increased difficulty in image registration phase as these manipulations may introduce more deviations in hash distance comparison. The examples in Fig. 8 illustrate the locations of tampered regions in all eight categories of tampered images with content-preserving manipulations ($D_{tampered\_cp}$). All tampered regions are correctly located.

C. Source Camera Identification

The proposed method is able to extract the source device (camera) information from the received hash data, hence the name "data-device hash". The receiver is able to validate that the received image is captured using a trusted device while detecting possible tampering and locate the modified content in the tampered image. Three cases are considered in order to prove the source camera identification capability. Case 1: for the same device, the distinguishability of the hash data produced by applying different PUF challenges. Case 2: for different devices, the distinguishability of hash data generated by applying the same challenge. These requirements are expected to be readily fulfilled by the hash generated through random PUF responses due to the inter-die variations of nano-scale CMOS device fabrication process. Last but not least, since the hash is dependent on both device and data information, it should have good anti-collision capability, which leads to Case 3: for the same device and same challenge, the distinguishability of the hash data produced by different images.

These desiderata are tested using eight CMOS image sensor based PUF instances and the test database of 240 authentic images. 10 Challenges are randomly selected and labelled as $c1 \sim c10$, while the eight PUF instances are labelled as $d1 \sim d8$. $c1$ and $d1$ are selected as the original challenge and device, respectively, to calculate the benchmarks hash values ($hash\_bm$) for the authentic images. In Case 1, a different challenge is applied to the same device to generate a new hash ($hash\_1$) for the same authentic image; For Case 2, the responses are collected from a different device using the same challenge set. The hash value ($hash\_2$) is then generated from these responses for the same authentic image; For Case 3, the 240 authentic images are re-ordered to make different original-received image pairs. This way, a different image is applied to generate a new hash ($hash\_3$) while the challenge and device remain unchanged. The authentication performance of each case is analyzed.

Each device and challenge combination is iterated on each image pair, there are altogether $8 \times 10 \times 240 = 19200$ tests for each case. Fig. 9(a) shows that changing the device key by either changing the challenge (Case 1) or the device (Case 2) will lead to 0 FAR in all malicious (device, challenge) pairs. The probability of collision of the hash generated by the same device key for two different images (Case 3) is very low, as evinced by the average FAR of only 0.000208 in Fig. 9(b). Though not ideal, the 0.000208 FAR for Case 3 is inconsequential. This is because those falsely accepted image pairs have irrelevant perceptual content with apparent semantic gap. For the use case of image forensics, they would have been rejected by visual inspection before being able to be presented as an evidence in a court of law. This is different from the use case of scanning large image databases for potential manipulations. The results show that the proposed method has good source camera identification performance for all three cases. Besides, it is noted that device fingerprint contributes more to differentiating the hash than the data (image). Therefore, introducing the device information into the hash increases the inter-class distance.

D. PUF Reliability Discussion

Albeit highly reliable, 100% correct regeneration of R and R_s is not guaranteed by the PUF. Since R_s is used as the seed of Knuth Shuffle algorithm, one bit flip can result in a completely different shuffle order. Bit errors of R_s can be corrected by Bose-Chaudhuri-Hocquenghen (BCH) [39] error correction code, which is highly flexible and hardware efficient. The reliability requirement of the much longer response R used for the random projection is fortunately not as strict as the short 128-bit R_s. To analyze the tolerance to bit errors of R, the unreliable R is created by randomly flipping some bits in the authentic responses while keeping R_s fully reliable by BCH error correction. The authentication performance is evaluated by injecting these unreliable responses into the proposed system. To minimize bias in the experimental results, the average acceptance rate is calculated for the unreliable R. Table III shows that an error rate of R in excess of 20% will definitely lead to an authentication failure even if a genuine image is presented. However, if the error rate of R is kept within 2%, the system can still maintain a very high correct detection rate of 99.8%. Fortunately, this 2% error rate is well

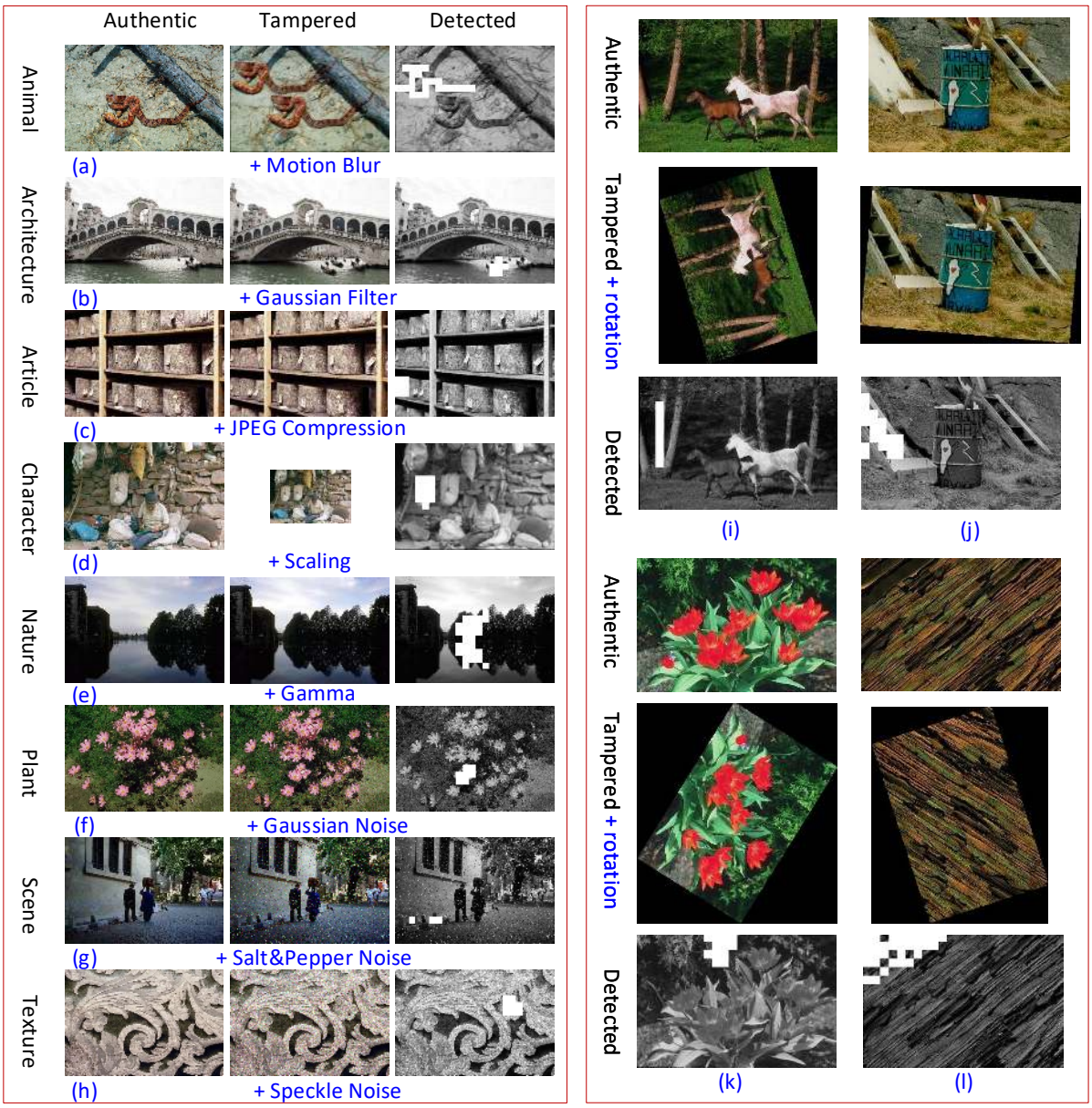

Fig. 8. Examples of tamper detection and location for all manipulations shown in Table II. (a)∼(h): Tampered image + content preserving manipulations including filtering, JPEG compression, scaling, Gamma correction and noising in all eight categories; (i)∼(l): Tampered image + rotation.

satisfied by the CMOS image sensor PUF over the industrial grade of operating temperature variation and typical regulated supply with no more 10% voltage variation. Moreover, the reliability of R can also be further enhanced by simpler majority voting technique.

TABLE III
SYSTEM PERFORMANCE UNDER UNRELIABLE R

error rate f	0.01	0.02	0.03	0.1	0.2	0.3∼1.0
accept rate	0.999	0.998	0.949	0.194	0	0

E. Security Analysis

A typical verification process of the proposed system involves (at least) an image to be validated, (at least) a claimed device and a verifier. The verifier determines if the image is captured by the claimed device without any malicious tampering. The trust model and the assumptions of the proposed system are given as follows:

Image: The received testing image may be either a genuine or maliciously tampered/replaced/fabricated version. It may have also gone through normal image processing including noising, filtering or geometrical transformations like rotation and scaling.

Device: The device has a monolithically integrated PUF, and data-device hash tag generation and comparison modules.

TABLE IV
COMPARISON WITH EXISTING WORKS.

Scheme	Perceptual robustness (parameters)		Tamper Detection	Tamper Location	Device Authentication
	Rotation	Scaling			
TIFS2012 [9]	0.9075 ($2^o \sim 30^o$)	0.8818 (0.5 $\sim$ 1.5)	Yes	Yes	No
TIFS2013 [40]	N/S (5^o)	N/S (0.5, 1.5)	Yes	Yes	No
TIFS2015 [10]	0.8609 (25°)	0.8477 (0.5)	Yes	Yes	No
TIFS2016 [11]	0.9926 ($\pm 1^\circ \sim \pm 90^\circ$)	1 (0.5 $\sim$ 2)	Yes	No	No
Springer2016 [2]	0.8002 ($\leqslant 5^\circ$)	1 (0.5 $\sim$ 1.5)	Yes	Yes	No
AsianHOST2016 [27]	No	No	Yes	No	Yes
ISCAS2018 [28]	No	No	Yes	Yes	Yes
This work	0.9625 (0 $\sim 360^\circ$)	0.9042 (0.5)	Yes (0.9542)	Yes	Yes

Noted that if several algorithms are proposed (e.g. [2], [9]), the one with best performance is chosen for comparison.

(a)

Case 1: Change of challenge

Case 2: Change of device

FAR	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10
d1	1	0	0	0	0	0	0	0	0	0
d2	0	0	0	0	0	0	0	0	0	0
d3	0	0	0	0	0	0	0	0	0	0
d4	0	0	0	0	0	0	0	0	0	0
d5	0	0	0	0	0	0	0	0	0	0
d6	0	0	0	0	0	0	0	0	0	0
d7	0	0	0	0	0	0	0	0	0	0
d8	0	0	0	0	0	0	0	0	0	0

Note: Original device: d1; Original challenge: c1; Image pairs: 240

(b)

	Case 3: Change of data (image)
Image pairs	240
#(device, challenge)	80
Average FAR	0.000208

Fig. 9. Source camera identification performance.

It is the claimed source camera of the image in question. The device may be either a genuine or maliciously tampered/replaced/fabricated version. Noted that any test circuit that has direct access to the challenge and response ports of the embedded PUF will be disabled or removed after testing so that the unobfuscated internal CRPs are inaccessible externally upon device deployment.

Verifier: The verifier is entrusted to verify the image content integrity and its claimed acquisition device. The verifier is assumed to be granted permission to use the device for this verification.

Attacker: The attacker is assumed to know all about the system except the CRPs of the trusted device. The adversary may try to deceive the verifier by sending fake images captured using an untrusted device or claiming the ownership of a stolen image. A common assumption is the adversary does not have the authorized device, and is unable to obtain the temporary data (such as intermediate results of the computations) stored in the volatile memory within the device while the latter is participating in an authentication process. In order to make the fake image/copyright pass the authentication, the adversary needs to recover the CRP mapping in order to generate a valid hash, which may possibly be achieved by device cloning, probing and random guessing.

Cloning attack refers to the duplication of another device that shares the same brand/type/function as the authenticated device in order to generate the device-dependent hash. However, PUF makes such attack infeasible even if the schematic, operation and other details of the camera and CMOS image sensor are made known to the attacker. Due to the uncontrollable manufacturing process variations, every device is unique and distinguishable even if the same mask set is used to refabricate the image sensor. As only the hash but not the native response bits are externally accessible, and the number of challenges is linearly proportional to the number of pixels (that make up the independent response bit-cells), it is impossible to machine learn the CRP mapping of the PUF by collecting the hashes from different input challenges. The PUF response is well obfuscated by the uniformly random shuffle, making its recovery from the hash data intractable.

Memory probing is an effective attack on traditional "secret-dependent" image hashes, where their secret keys are locally stored in a NVM. If the secret key has been successfully retrieved, the adversary can easily generate a valid hash for any image he/she has stolen or forged. Storing secret key in NVM has been found to be vulnerable to invasive attacks like reverse-engineering. In our case, the secret used to generate the hash is not stored but directly built into the device structure (CMOS image sensor array) as an integral property of the hash function, and can only be generated upon request when the device is powered on. Any invasive or semi-invasive attacks on the CMOS image sensor chip will easily damage the device structure and erase the secret permanently. Hence, probing attacks are unlikely to succeed.

Random guessing is another common attack. For an adversary who wants a forged image to be authenticated without the correct device, he/she may try to create an effective hash by trial and error. Noted that an effective hash generation requires both valid random projection response R and shuffle response R_s, among which R_s is of paramount importance. After correctly cracking R_s, the attacker may generate a valid internal hash H_o for the forged image using the same challenge either by performing the lunch time attack as mentioned in the last two paragraphs of Sec. III-B or directly guessing R. However, the probability of successfully cracking a 128-bit R_s is only 2^{-128}, which can be made even more negligible by increasing the bit-length of R_s. Without the correct R_s, the original hash H_o cannot be correctly recovered, which makes conducting further lunch time attack impossible. As for the random guessing of R, Table III shows that on the premise of correctly regenerated R_s, the acceptance rate for a genuine

image is merely 19.4% even if the adversary makes only 10% of errors in guessing R. Assume that 0 and 1 bits are equally probable to occur in the PUF response, the probability P_r of making at most f fraction of bit errors in an N-bit response by random guessing is given by:

$$P_r = \sum_{i=\lceil Nf' \rceil}^{N} \binom{N}{i} (\frac{1}{2})^i (\frac{1}{2})^{N-i} = 2^{-N} \sum_{i=\lceil Nf' \rceil}^{N} \binom{N}{i} \quad (25)$$

where $f' = 1 - f$. When $N = 2048$ and $f = 0.1$, $P_r < 2.45 \times 10^{-558}$. This implies that it is close to impossible for an adversary to gain even 19.4% authentication accuracy by random guessing, even with the knowledge of R_s, let alone it is also nearly impossible to recover R_s.

F. Comparison

In this section, the proposed work is compared with the existing perceptual image hashing methods in recent years. The comparison is made from four main perspectives: perceptual robustness and the capabilities of tamper detection, tamper region location and device authentication. For perceptual robustness, different non-uniform content-preserving manipulations as well as their parameters were used in different works, which make the comparison difficult and possibly unfair. It is noticed that for content-preserving manipulations like noising, filtering, JPEG compression and gamma correction, most of the perceptual hashing methods can achieve good performance ($\geqslant 0.95$). However, perceptual robustness against geometric transformations like rotation and scaling is a widely discussed key challenge in perceptual hashing research. Based on these observations, only "Rotation" and "Scaling" are listed under perceptual robustness in Table IV for comparison. Table IV compares the performance of our proposed work based on a single manipulation of the operations in Table I against the state-of-the-art perceptual image hashing methods. The experiment results show that our proposed work is the only work that can achieve tamper detection, tamper region location and device authentication while maintaining a high perceptual robustness against rotation and scaling. Noted that though the true positive rates for "Rotation" and "Scaling" are not as high as those of [11], the proposed method achieves a TDR of 0.9542 while the TDR of [11] is not given. The trade-off between perceptual robustness and TDR is inevitable. Since this work targets digital image forensics, the parameters are skewed in favor of tamper detection. It is acceptable to have a small sacrifice on perceptual robustness to trade for better tamper detection performance.

VI. Conclusion

As the first robust rotation-/scaling-invariant PUF-based perceptual image hash, the proposed data-device hash has introduced an added attribute of birth certification, which is essential in digital forensics to prove the authenticity of a visual evidence conveyed by the image content. This is made possible using the idea of random projection, i.e., projecting the content-based image features into a CMOS image sensor PUF defined Bernoulli random matrix. The proposed hash carries both time, data and device information. Not only can it detect and precisely locate image forgeries, but also identify the camera of the source image with high accuracy. Besides, the proposed hash is robust against normal content-preserving manipulations such as noising, filtering, JPEG compression, Gamma correction, rotation, scaling, etc. The proposed work is more secure than existing image hashes that rely on a locally stored secret key for the generation and validation, as the random space used for mapping the feature points is generated only on demand by a tamper-resistant PUF. Invasive or semi-invasive attacks on the device to recover the CRP mapping will produce unpredictable fault in the hash generation. Forging a hash by random guessing of PUF response has been calculated to be almost impossible.

References

[1] S. Cole, "Reddit just shut down the deepfakes subreddit," Feb. 2018. [Online]. Available: https://motherboard.vice.com/en_us/article/neqb98/reddit-shuts-down-deepfakes

[2] R. Davarzani, S. Mozaffari, and K. Yaghmaie, "Perceptual image hashing using center-symmetric local binary patterns," *Multimedia Tools Appl.*, vol. 75, no. 8, pp. 4639–4667, Mar. 2016.

[3] Y. Guo, O. C. Au, R. Wang, L. Fang, and X. Cao, "Halftone image watermarking by content aware double-sided embedding error diffusion," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3387–3402, Jul. 2018.

[4] X. L. Liu, C. C. Lin, and S. M. Yuan, "Blind dual watermarking for color images authentication and copyright protection," *IEEE Trans. Circuits Syst. Video Techno.*, vol. 28, no. 5, pp. 1047–1055, May 2018.

[5] T. H. Thai, R. Cogranne, F. Retraint *et al.*, "JPEG quantization step estimation and its applications to digital image forensics," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 1, pp. 123–133, Jan. 2017.

[6] L. Wen, H. Qi, and S. Lyu, "Contrast enhancement estimation for digital image forensics," *ACM Trans. Multimedia Comput., Commun., Appl. (TOMM)*, vol. 14, no. 2, p. 49, May 2018.

[7] P. Korus and J. Huang, "Improved tampering localization in digital image forensics based on maximal entropy random walk," *IEEE Signal Process. Letters*, vol. 23, no. 1, pp. 169–173, Dec. 2016.

[8] J. A. Redi, W. Taktak, and J.-L. Dugelay, "Digital image forensics: a booklet for beginners," *Multimedia Tools Appl.*, vol. 51, no. 1, pp. 133–162, 2011.

[9] X. Lv and Z. J. Wang, "Perceptual image hashing based on shape contexts and local feature points," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 1081–1093, Jun. 2012.

[10] X. Wang *et al.*, "A visual model-based perceptual image hash for content authentication," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 7, pp. 1336–1349, Jul. 2015.

[11] Z. Tang, X. Zhang, X. Li, and S. Zhang, "Robust image hashing with ring partition and invariant vector distance," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 1, pp. 200–214, Jan 2016.

[12] P. Meerwald, C. Koidl, and A. Uhl, "Attack on watermarking method based on significant difference of wavelet coefficient quantization," *IEEE Trans. Multimedia*, vol. 11, no. 5, pp. 1037–1041, Aug. 2009.

[13] T. Qazi *et al.*, "Survey on blind image forgery detection," *IET Image Process.*, vol. 7, no. 7, pp. 660–670, Oct. 2013.

[14] S. E. Quadir *et al.*, "A survey on chip to system reverse engineering," *ACM J. Emerging Technol. Comput. Syst. (JETC)*, vol. 13, no. 1, p. 6, Dec. 2016.

[15] J. Breier, D. Jap, and C. N. Chen, "Laser profiling for the back-side fault attacks: with a practical laser skip instruction attack on AES," in *Proc. the 1st ACM Workshop Cyber-Physical Syst. Security*, Singapore, Apr. 2015, pp. 99–103.

[16] C. Cimpanu, "Security flaw lets attackers recover private keys from Qualcomm chips," Apr. 2019. [Online]. Available: https://www.zdnet.com/article/security-flaw-lets-attackers-recover-private-keys-from-qualcomm-chips/

[17] B. Xu, X. Wang, X. Zhou, J. Xi, and S. Wang, "Source camera identification from image texture features," *Neurocomputing*, vol. 207, pp. 131–140, Sept. 2016.

[18] V. U. Sameer, A. Sarkar, and R. Naskar, "Source camera identification model: Classifier learning, role of learning curves and their interpretation," in *Proc. 2017 Int. Conf. Wireless Commun., Signal Process. Networking (WiSPNET)*, Chennai, India, Mar. 2017, pp. 2660–2666.

[19] A. Roy, R. S. Chakraborty, V. U. Sameer, and R. Naskar, "Camera source identification using discrete cosine transform residue features and ensemble classifier." in *Proc. 2017 IEEE Conf. Comput. Vision Pattern Recognition Workshops (CVPRW)*, Honolulu,USA, Jul. 2017, pp. 1848–1854.

[20] J. Lukáš, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 2, pp. 205–214, Jun. 2006.

[21] M. Chen, J. Fridrich, M. Goljan, and J. Lukás, "Determining image origin and integrity using sensor noise," *IEEE Trans. Inf. Forensics security*, vol. 3, no. 1, pp. 74–90, Mar. 2008.

[22] D. Valsesia, G. Coluccia, T. Bianchi, and E. Magli, "Large-scale image retrieval based on compressed camera identification," *IEEE Trans. Multimedia*, vol. 17, no. 9, pp. 1439–1449, Sept. 2015.

[23] D. Valsesia, G. Coluccia, T. Bianchi *et al.*, "User authentication via PRNU-based physical unclonable functions," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 8, pp. 1941–1956, Aug. 2017.

[24] A. Lawgaly and F. Khelifi, "Sensor pattern noise estimation based on improved locally adaptive DCT filtering and weighted averaging for source camera identification and verification," *IEEE Trans. Inf. Forensics and Security*, vol. 12, no. 2, pp. 392–404, Feb. 2017.

[25] Y. Zheng, Y. Cao, and C. H. Chang, "A new event-driven dynamic vision sensor based physical unclonable function for camera authentication in reactive monitoring system," in *Proc. IEEE Asian Hardware-Oriented Security Trust (AsianHOST)*, Yilan,Taiwan, Dec. 2016, pp. 1–6.

[26] Y. Cao, L. Zhang, S. S. Zalivaka, C. H. Chang, and S. Chen, "CMOS image sensor based physical unclonable function for coherent sensor-level authentication," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 11, pp. 2629–2640, Nov. 2015.

[27] Y. Cao, L. Zhang, and C. H. Chang, "Using image sensor PUF as root of trust for birthmarking of perceptual image hash," in *Proc. IEEE Asian Hardware-Oriented Security Trust (AsianHOST)*, Yilan,Taiwan, Dec. 2016, pp. 1–6.

[28] Y. Zheng, S. S. Dhabu, and C. H. Chang, "Securing IoT monitoring device using PUF and physical layer authentication," in *Proc. 2018 IEEE Int. Symp. Circuits Syst. (ISCAS)*, Florence, Italy, May 2018, pp. 1–5.

[29] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. vision image understanding*, vol. 110, no. 3, pp. 346–359, Jun. 2008.

[30] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. 2001 IEEE Comput. Soc. Conf. Comput. Vision. Pattern Recognition*, vol. 1, Kauai, HI, USA, Apr. 2001, pp. I–I.

[31] E. Oyallon and J. Rabin, "An analysis of the SURF method," *Image Proc. Line*, vol. 5, pp. 176–218, May 2015.

[32] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. comput. vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[33] M. Zuliani, "RANSAC for dummies," Vision Research Lab, Uni. California, Santa Barbara, Jul. 2014. [Online]. Available: http://www.cs.tau.ac.il/ turkel/imagepapers/RANSAC4Dummies.pdf

[34] C. H. Chang, Y. Zheng, and L. Zhang, "A retrospective and a look forward: Fifteen years of physical unclonable function advancement," *IEEE Circuits Syst. Mag.*, vol. 17, no. 3, pp. 32–62, Aug. 2017.

[35] P. Preeti and K. S. Rajeev, "A survey: digital image watermarking techniques," *Int. J. Signal Process. Image Process. Pattern Recognition*, vol. 7, no. 6, pp. 111–124, 2014.

[36] S. S. Vempala, *The Random Projection Method*. American Mathematical Soc., 2005, vol. DIMACS-65.

[37] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, "The Johnson-Lindenstrauss lemma meets compressed sensing," *preprint*, vol. 100, no. 1, pp. 1–9, Jan. 2006.

[38] J. Dong, W. Wang, and T. Tan, "CASIA image tampering detection evaluation database," in *Proc. 2013 IEEE China Summit Int. Conf. Signal Inf. Process. (ChinaSIP)*, Beijing,China, Jul. 2013, pp. 422–426.

[39] S. S. Zalivaka, L. Zhang, V. P. Klybik, A. A. Ivaniuk, and C.-H. Chang, "Design and implementation of high-quality physical unclonable functions for hardware-oriented cryptography," in *Secure System Design and Trustable Computing*. Springer, 2016, pp. 39–81.

[40] Y. Zhao, S. Wang, X. Zhang, and H. Yao, "Robust hashing for image authentication using zernike moments and local features," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 1, pp. 55–63, Jan. 2013.

Yue Zheng (S'15) received the B. Eng. degree from Shanghai University (SHU), China in 2015. She is currently working towards the Ph.D. degree in School of Electrical and Electronic Engineering, Nanyang Technological University (NTU) Singapore. Her area of research includes hardware security, physical unclonable function for device fingerprinting and secret key generation. She was an exchange student with Kyoto University in 2019.

Yuan Cao (S'09-M'14) received his B.S. degree from Nanjing University, MPhil. degree from Hong Kong University of Science and Technology and Ph.D. degree from Nanyang Technological University in 2008, 2010 and 2015, respectively. Currently he works as a professor in College of Internet of Things Engineering of Hohai University. His research interests include hardware security, silicon physical unclonable function, and analog/mixed signal VLSI circuits and systems.

Chip-Hong Chang (S'92-M'98-SM'03-F'18) received the B.Eng. (Hons.) degree from the National University of Singapore in 1989, and the M. Eng. and Ph.D. degrees from Nanyang Technological University (NTU) of Singapore in 1993 and 1998, respectively. He is an Associate Professor of the School of Electrical and Electronic Engineering (EEE) of NTU. He held joint appointments with the university as Assistant Chair of Alumni from 2008 to 2014, Deputy Director of the Center for High Performance Embedded Systems from 2000 to 2011, and Program Director of the Center for Integrated Circuits and Systems from 2003 to 2009. He has coedited 4 books, published 10 book chapters, 100 international journal papers (two-thirds are IEEE) and more than 170 refereed international conference papers (mostly in IEEE), and delivered over 40 colloquia. His current research interests include hardware security, unconventional number systems, and low-power and fault-tolerant digital signal processing algorithms and architectures.

Dr. Chang serves as the Associate Editor of IEEE Transactions on Very Large Scale Integration (VLSI) Systems since 2011, IEEE Access since 2013, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems and IEEE Transactions on Information Forensics and Security since January 2016, IEEE Transactions on Circuits and Systems-I from 2010-2013, Integration, the VLSI Journal from 2013-2015, Springer Journal of Hardware and System Security since June 2016 and Microelectronics Journal since May 2014. He guest edited several journal special issues and served in the organizing and technical program committee for more than 60 international conferences. He is also an IET Fellow and 2018-2019 Distinguished Lecturer of IEEE Circuits and Systems Society.