UDhashing : physical unclonable function-based user-device hash for endpoint authentication

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

2019

Zheng, Y., Cao, Y., & Chang, C.-H. (2019). UDhashing : physical unclonable function-based user-device hash for endpoint authentication. IEEE Transactions on Industrial Electronics, 66(12), 9559-9570. doi:10.1109/TIE.2019.2893831

<https://hdl.handle.net/10356/145813>

<https://doi.org/10.1109/TIE.2019.2893831>

© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at:
<https://doi.org/10.1109/TIE.2019.2893831>

Downloaded on 26 Apr 2023 18:40:11 SGT

UDhashing: Physical Unclonable Function Based User-Device Hash for Endpoint Authentication

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

Abstract—With IT consumerization, access control to remote system by endpoint user and endpoint device is mandatory for security and privacy protection. Existing systems bind an end-user with his/her registered devices but authenticate only the user and device independently. This work presents a novel UDhashing scheme, which is capable of providing a bipartite authentication of both end-user and end-device as a whole, and mutual authentication between the endpoint and the verifier. Non-contact facial biometric is extracted as user identity and physical unclonable function (PUF) is embedded into the device to generate a device “fingerprint”. UDhashing serves as an intermediary to unify the macroscopic human biometric and microscopic silicon entropy source into a single identity. The scheme is demonstrated using measured silicon data of a diode-clamped inverter based strong PUF fabricated in 40nm 1.1V CMOS technology, and the ORL and Ext. Yale B face databases. The experimental results show that the proposed system has good authentication performance with excellent discriminability for different (challenge, user, device) tuples. Besides, the proposed system is analyzed to be resilient to several known attacks. Its reliability and authentication performance can be easily enhanced by low-cost error-correction technique without compromising security.

Index Terms—Authentication, Cancelable Biometrics, Physical Unclonable Function, Security.

I. INTRODUCTION

AS the consumerization of IT grows, increasing number of network environments are faced with frequent influx of external or remote users using personal devices. Such trend thrives with Bring Your Own Device (BYOD) policy and online gaming. BYOD system allows the employees to connect their personally hold devices to the inner corporate network without the restriction of time and place. This developing phenomenon in enterprise IT is irresistible with appealing productivity boost and cost saving. Its security, however, is also of big concern. Unauthenticated access to sensitive

information from external or remote personal devices makes the whole system vulnerable to network attacks. In game consoles like PS4, one Play Station Network account is only allowed to bind with one game console as admin user. Such scheme can control the number of shared online accounts to protect the profits of game providers, but secure authentication between endpoint users and endpoint devices is still lacking or inadequate. Though named differently, configuration managers like “End-User Device Registration” by VMware [1], “User Device affinity” by Microsoft [2], and “User Device Tracker” by SolarWinds [3] have been established with a common aim to support commercial remote centers to automatically link each authorized end-user with one or several legitimated end-devices in order to detect and deny access when rogue devices and users are trying to connect to the network.

There are several design requirements for a robust endpoint authentication system. Firstly, since the objective of IT consumerization is to improve work efficiency and user experience, easy and fast log-in methods like a glance, a touch or an utter of speech are preferred to hard to remember long password. Secondly, the system should have high security level. Storing any secret credentials such as password or encryption key is unsafe as digital content in non-volatile storage is known to be recoverable by invasive and semi-invasive attacks [4]. Besides, user privacy needs to be protected to refrain the endpoint from giving out sensitive information to masquerade system.

Biometrics like face, fingerprint, iris, etc. are ideal candidates for user identification due to their many appealing attributes such as easy to install and use, non-repudiable accountability, security, cost, etc. The strong and permanent link between human and their biometrics also creates great concern on the preservation of biometric templates. Directly storing the biometrics in the template dictionary is vulnerable to attacks. Once a template is stolen or compromised, it is impossible for the individual to reproduce a new biometric pattern, which leads to privacy leakage and security collapse of other systems that use the same type of biometric. Existing countermeasures [5], [6] include storing biometric templates in secure hardware like A8 chip on Apple iPhone6, encrypting them using encryption algorithms like AES and RSA, or adopting feature transformations like biohashing. As these countermeasures identify only the user but not the device, lost or stolen device put registered biometric templates at great risk. Confidential information may be accessed with proxy

Manuscript received June 27, 2018; revised September 28, 2018 and November 15, 2018; accepted January 3, 2019; This project was supported by the Singapore Ministry of Education MOE Tier 1 grant No. 2018-T1-001-131, RG87/18 (S), and the National Natural Science Foundation of China (61601168).

Y. Zheng and C. H. Chang (corresponding author, email: echchang@ntu.edu.sg) are with the School of Electrical and Electronic Engineering, Nanyang Technological University, 639798, Singapore. Y. Cao is with the Hohai University, Changzhou, China.

device by forging the user biometric.

In much the same way as biometric is used to identify different users, each device can also be identified by its device-unique “fingerprint”, which can be generated by a physical unclonable function (PUF) [4]. A PUF is a hardware security primitive that exploits random unpredictable manufacturing process variations of identically designed circuits to generate a unique response upon challenged by a query. As PUF does not rely on secure storage of any secrets like conventional hardcoded electronic tag, digital signature or cryptographic hash, it holds strong promise to protect the biometric templates for bipartite authentication, i.e., to identify the user based on his/her personal device and the device based on its associated owner.

An intuitive way to exploit human biometric and PUF for end-user and end-device identification is by simple two-factor authentication, where the user and device are independently identified by the biometric and PUF response. Only when both authentications are successful will the request be granted. However, the attack difficulty for this layered solution is only an aggregation of each single layer of protection. Moreover, this does not address the additional security needed for protecting the extracted biometric templates. A seamless and tighter assimilation of biometric and PUF is required. The unified identity is a unique bitstream inseparably derived from the randomness extracted from a physical device and human biometric features. Only a handful of attempts have fulfilled this “UD PUF” notion [7]–[9]. The key idea behind [7], [8] is to differentiate different users and devices by extracting raw biometric information like touch screen pressure or voice with inherited silicon variations of the sensor. As the desirable intrinsic parametric changes contributed by device fabrication process variations are not structurally harnessed in these methods, they can be polluted by other systematic or random noises, resulting in high identification error rate for the (same user, same challenge, different device) combination. Amplifying the parametric deviations to improve device identification will reduce the sensitivity of user biometric whereas noise reduction in biometric information processing will distort device parametric distribution. The very recent work [9] proposed a *BLOcKeR* framework for match-on-card/device applications. The basic idea is to generate an obfuscated biometric template (*obs_key*) by feeding the processed biometric feature into a strong PUF as the challenge. However, this work has strict requirements on the extracted biometric features, which are usually noisy in practice. Due to the avalanche effect of ideal PUF, a small change in the challenge will lead to dramatic bit flips in the response. As the quantized biometric feature-based challenge has to be 100% accurate to ensure that the same *obs_key* will be generated in the authentication phase, a NA-IOMBA scheme is proposed to select the most robust biometric feature for each individual user. Accurate noise samples/models over time and different conditions need to be provided to ensure its performance. Besides, the scheme can only generate one determinant *obs_key* for a (user, device) combination. Once *obs_key* is leaked, the security of using the particular device is compromised.

In this paper, a “UDhashing” scheme is proposed to resolve

this dilemma and overcome a potential security weakness in our preliminary proposal [10]. To bind a device to its user, a secret key is derived from its embedded PUF to transform its user live biometric into a unique hashed pattern. A successful authentication will require both a correct biometric input of a user into his/her registered device and a correct response to a query from that device. A hashed biometric pattern is allowed to be revoked or reissued to achieve cancelable biometric. Unlike our preliminary work [10], no secret information will be leaked in communication as the unification of user and device features into bio-code is performed at the end-device while the authentication is conducted by the server. An improved communication protocol is also proposed to achieve mutual authentication. In contrary to existing user-device authentication solutions [7], [8], [10], both user and device identities are structurally and independently derived before they are fused. Since only the extracted feature in hashed domain is transmitted, the system is resource efficient in protecting user privacy against lost/tampered/stolen biometric templates. As opposed to the “match-on-device” scenario [9], our proposed “match-on-server” authentication scheme eases the biometric feature extraction process, and allows a (user, device) combination to be refreshed by applying a different challenge.

II. RELATED WORKS

A. Cancelable Biometrics based on Random Projection

Random projection is widely used for dimension reduction. The key idea stems from Johnson-Lindenstrauss Lemma [11], which states that a set of M points in a high-dimensional space can be mapped into a space of much lower dimension ($O(\epsilon^{-2} \log M)$) such that the pair-wise distances are approximately preserved:

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

where $f()$ is the projection function and ϵ is an arbitrary fraction between 0 and 1. The above property has been demonstrated to be true when the projection matrix R is a **random orthonormal matrix**, i.e., its columns are orthogonal unit vectors [12].

Leveraging on this distance-preserving property, random projection has been introduced into cancelable biometrics [6], [13]–[15], a.k.a. biohashing. By projecting the raw biometric features into a user-specific random space, the final generated bio-code is a mutable transformation of the user’s biometric features across different databases. The transformed templates are transmitted or stored for later processing. The loss/theft/tampering of these templates will not result in permanent compromise of their users’ biometric, hence the cancelability. The basic steps of biohashing are described below [13], [15]:

- Apply feature extraction to the raw biometric data to obtain an n -bit invariant feature vector y .
- Generate a pseudo-random matrix of size $n \times m$ using a user-specific token: $\{r_i \in \mathbb{R}^n | i = 1, 2, \dots, m\}$, where $m < n$ is the length of the final bio-code.
- Apply the modified Gram-Schmidt (mGS) algorithm to transform the random matrix r into an orthonormal matrix

$\{R_i \in \mathbb{R}^n | i = 1, 2, \dots, m\}$. mGS consistently produces more orthogonal vectors than the classical GS. Its loss of orthogonality¹ depends only linearly on the condition number of original dataset and fails completely when r is singular [16], [17].

- Make a random projection of y with R via the inner product:

$$v = \langle R, y \rangle = R^T y \quad (2)$$

where v is the projected vector of the reduced dimension. Since $m < n$ for the transposed orthonormal matrix R^T (size: $m \times n$), it is impossible to find the exact value of y by solving this underdetermined system of linear equations under the premise that the number of possible solutions is infinite.

- The m -bit digitized bio-code $\{b_i | i = 1, 2, \dots, m\}$ is obtained by comparing v_i with a pre-defined threshold τ , i.e.,

$$b_i = \begin{cases} 0 & \text{if } v_i \leq \tau \\ 1 & \text{if } v_i > \tau \end{cases} \quad (3)$$

The quantization threshold τ in (3) is selected to have approximately equal probability of generating each binary digit(s) ('0' or '1' for one-bit quantization, '00', '01', '10' or '11' for two-bit quantization, etc.) to maximize the entropy of the m -bit bio-code. Hence, τ can be determined from the mean of the distribution of v during the enrollment phase.

Given two feature vectors, y_1 and y_2 , and a random space R , the Euclidean distance of transformed features is given by:

$$\begin{aligned} \|v_1 - v_2\|^2 &= \langle v_1 - v_2, v_1 - v_2 \rangle \\ &= \langle v_1, v_1 \rangle - 2\langle v_1, v_2 \rangle + \langle v_2, v_2 \rangle \\ &= (R^T y_1)^T R^T y_1 - 2(R^T y_1)^T R^T y_2 + (R^T y_2)^T R^T y_2 \quad (4) \\ &= y_1^T R R^T y_1 - 2y_1^T R R^T y_2 + y_2^T R R^T y_2 \\ &\xrightarrow{R R^T = I} y_1^T y_1 - 2y_1^T y_2 + y_2^T y_2 = \|y_1 - y_2\|^2 \end{aligned}$$

Eq. (4) is true only if $R R^T = I$, where I is an identity matrix. However, the non-square matrix R used in biohashing applications is a semi-orthogonal matrix of dimension $n \times m$ ($n > m$), which means that only the columns of R are orthonormal vectors after mGS. Therefore, $R R^T$ is not exactly equal to I , but decomposable into $R R^T = I + \delta_{i,j}$. Increasing m will provide better approximation of $R R^T \approx I$, and $\delta_{i,j}$ approaches 0 asymptotically when $n \rightarrow m$.

B. Constraints of Existing Biohashing Scheme

Despite the ease of implementation, security of existing biohashing schemes can be undermined by some hidden caveats. The random number used to specify the projection is assumed to be identical for the same user in the same application, and should be uncorrelated and distinguishable across users and applications to prevent cross matching. This random number is currently generated via a tokenized pseudo-random number (TPRN), which requires a seed to be safely kept in a token device or smart card [18]. Reported performances of Biohashing schemes are achieved with the assumption that the

TPRN is not compromised, shared or cloned. This assumption may not hold as the TPRN suffers from the same vulnerability as the "token", i.e., its seed has to be stored in a physical device. Secondly, once the transformed biometric template has been lost, tampered or stolen, the system should in principle be capable of revoking or issuing a new template easily but this capability may be crippled if the victim is the TPRN. Last but not least, there is a trade-off between authentication performance and template security when a cancelable bio-code is used. The offset-tolerance required to maintain accurate identification of a correct user-device pair cannot be fulfilled efficiently by existing user-device independent biohashing schemes.

III. PROPOSED CANCELABLE UDHASHING SCHEME

To provide an efficient yet computationally feasible solution for endpoint authentication, we propose to unify two innate identities derived from the user and the device independently. The introduction of device biometric not only fulfills the user device affinity but also mitigates all the constraints discussed in Sec. II-B in applying existing biohashing for this purpose.

A. Facial Feature Representation

In our preliminary work [10], the user biometric and device signature are unified on the server side, which requires the biometric characteristics (face) and the raw device key (PUF response) to be sent to the server for hashing. However, the transmission of sensitive data over binary channel is vulnerable to man-in-the-middle (MITM) attacks, and incurs redundant data transmission. Since the proposed system is designed to authenticate remote end-user and end-device, the onus of authentication is on the server as it is much harder and resource inordinate to secure the template and trained model in easily accessible local device. To implement the UDhashing module on an end-device, the biometric features are required to be efficiently computed before projecting into a device-specific space. The previously adopted factorial methods in high-dimensional spaces like PCA and LDA are not suitable due to their computational complexity. Instead, Gabor filters and Discrete Cosine Transform (DCT) are used for feature extraction. Sparse representation based classification is applied for recognition.

1) *Gabor Features*: Gabor filter is robust to illumination variations. It looks for specific frequency content in specific directions around the region of analysis of an image. In the spatial domain, a 2D Gabor filter $\varphi_{\mu,\nu}(x, y)$ is a Gaussian kernel function modulated by a sinusoidal plane wave whose center frequency and orientation are given by f_μ and θ_ν , respectively [19], i.e.,

$$\varphi_{\mu,\nu}(x, y) = \frac{f_\mu^2}{\pi k \eta} e^{-((f_\mu^2/k^2)x'^2 + (f_\mu^2/\eta^2)y'^2)} e^{j2\pi f_\mu x'} \quad (5)$$

where $x' = x \cos \theta_\nu + y \sin \theta_\nu$, $y' = -x \sin \theta_\nu + y \cos \theta_\nu$, $f_\mu = f_{max}/2^{(\mu/2)}$, and $\theta_\nu = \nu\pi/8$.

To obtain the Gabor features, a set of Gabor filters $\varphi_{\mu,\nu}(x, y)$ is constructed, which typically has a size of five scales ($\mu = 0, 1, \dots, 4$) and eight orientations ($\nu = 0, 1,$

¹Based on the logarithm of the condition number of ~ 1.3 for the 320×100 matrix r generated by our PUF, the loss of orthogonality is kept below 10^{-14} according to [16].

Fig. 1. (a) An example of DCT-II feature extraction for a 4 × 4 image subblock of I ; (b) The proposed Cancelable UDhashing scheme; (c) The proposed system overview.

$\dots, 7)$ [19]. For a given face image $I(x, y)$, its Gabor feature corresponding to the parameter set (μ, ν) can be calculated by

$$G_{\mu, \nu}(x, y) = I(x, y) * \varphi_{\mu, \nu}(x, y) \quad (6)$$

where $G_{\mu, \nu}(x, y)$ is the complex filtering output, whose magnitude is used for face representation. However, the augmented Gabor (magnitude) features are usually very large. For example, a simple image of size 128×128 will result in a $128 \times 128 \times 40 = 655360$ -bit feature vector after filtering with 40 Gabor filters. Downsampling is needed to further reduce its dimensionality for efficient data processing and storage.

2) *DCT-II Features*: Type II DCT is most commonly used to transform spatial information to decoupled frequency information for local feature extraction. It has the least flop count for unitary normalization and can be efficiently computed by pruning redundant operations from a real-even discrete Fourier transform. Discriminable details between images are mainly bunched in the *DC* coefficient and the first few dominant *AC* DCT coefficients. The insignificant *AC* coefficients of small values can be removed with negligible impact on offset tolerance. To extract the features, an image I is usually divided into several blocks, and then DCT-II is applied to each block. As illustrated in Fig. 1(a), the first N DCT coefficients of each block are extracted in a raster scan order, and then concatenated together as the local features [20].

B. Device Key Generation

The unique device signature is provided by the embedded PUF, which produces a sequence of unpredictable random response bits only on demand. Strong PUF with a practically inexhaustible number of Challenge-Response Pairs (CRPs) is preferred for authentication. As each CRP will be used only once for each authentication session for the same user-device group, replay attack is impossible and correlations between the system responses cannot be exploited to predict the response or target a small subset of used responses of the PUF to figure out the unused CRPs enrolled in the server.

To integrate PUF into the authentication scheme, the PUF response can either be loaded as a seed of the pseudo-random number generator (PRNG) or used directly as a random matrix for projection. As shown in Sec. II, the random matrix has a size of $n \times m$. To relax the length of random bits required to be generated by the PUF, the PUF response can be used as a seed to a PRNG. However, environmental variations and device aging may result in response bit flip in PUF. Due to the avalanche effect of PRNG, one bit error in the seed value will produce multiple bit errors. Consequently, a completely different random matrix will be used in the authentication phase from that in the enrollment phase, which will result in authentication failure. Therefore, error correction code (ECC) is needed to reconcile the potential response bit error generated by the PUF under varying operating environments. Using the PUF response bits to construct the random matrix directly will minimize authentication failures due to the reliability of PUF. A small fraction of bit errors in PUF responses is tolerable since a similarity score is calculated for the classification in authentication. Using a strong PUF enables a large number of response bits to be generated for the random matrix. Besides, simple majority voting can be used to further reduce the bit error rate (BER) of PUF to eliminate the false negatives.

Two important factors need to be considered in selecting a strong PUF for endpoint user-device authentication. Firstly, the PUF should have good randomness and uniqueness. High uniqueness is important to ensure that a unique random matrix is generated for every different challenges and different devices. A good randomness makes the random matrix full rank and difficult to guess or predict by unauthorized users. Secondly, strong PUFs can be vulnerable to modeling attacks [4], [21]. It is important to prevent the projected bio-codes from being attacked by the successful modeling of strong PUF. For these reasons, the prototype is designed with a diode-clamped inverter based strong PUF (monoPUF) [22], which is derived from a nonlinear structure and has been demonstrated to have good randomness and uniqueness, as well as high

reliability and resistance to modeling attacks.

C. Bio-code Generation and Recognition

The feature dimension obtained after applying Gabor filters and DCT is usually very large. For an image I of size 128×128 , the dimension of Gabor features extracted by a set of Gabor filters with 5 scales and 8 orientations is around $128 \times 128 \times 5 \times 8 \div 256 = 2560$ even with a downsampling rate of 256. By keeping the first 100 coefficients for each subblock after applying DCT-II with block size of 32×32 , the DCT feature dimension is reduced to $(128 \div 32)^2 \times 100 = 1600$. The size of the concatenated features of $2560 + 1600 = 4160$ is still large. Increasing the downsampling rate in Gabor feature extraction or reducing the number of selected coefficients in each DCT-II block can help to reduce the feature size, but it will also degrade the feature quality, leading to unsatisfactory recognition results for later processing. As mentioned in Sec. III-B, the high dimension of biometric feature and satisfaction of (1) require a proportionally large random sequence for the mGS algorithm. Even if strong PUF is adopted for the direct generation of random matrix, for cost effectiveness, the concatenated features are segmented, and random projection is applied to each feature segment.

Fig. 1(b) shows an overview of the proposed UDhashing scheme. Instead of using a TPRN in biohashing, PUF responses are generated to build the projection space. Majority voting is applied before mGS algorithm to enhance the reliability of PUF response defined device-specific random matrix. The biometric feature vector is uniformly divided into t segments. The segmented features are then projected into the device-unique space correspondingly. The transformed feature patterns for all segments are concatenated to form a transformed template, which is digitized to obtain the UDhashing bio-code.

Sparse Representation based Classification (SRC) is implemented on the server side for (challenge, user, device) tuple recognition. It is comparatively computational expensive to be implemented in the resource-constrained end-devices. A simple procedure of using SRC for this tuple recognition is described as follows [14]:

Suppose we have L distinct (challenge, user, device) classes, each with s m -dimensional training bio-codes x obtained in the enrollment phase. An overcomplete dictionary D is constructed by simply concatenating these bio-codes.

$$D = [D_1, D_2, \dots, D_L] \in \mathbb{R}^{m \times (s, L)} \quad (7)$$

$$= [x_{11}, \dots, x_{1s} | x_{21}, \dots, x_{2s} | \dots | x_{L1}, \dots, x_{Ls}]$$

Given sufficient training samples of each tuple class, any unknown test sample belonging to the same class will lie approximately in the linear span of the training samples. Let x be a bio-code test sample of class k , then its corresponding observation vector $y \in \mathbb{R}^m$ is a linear combination of the training bio-codes, i.e.,

$$y = \sum_{i=1}^L \sum_{j=1}^s \alpha_{ij} x_{ij} = D\alpha \quad (8)$$

where $\alpha = [\alpha_{11}, \dots, \alpha_{1s} | \alpha_{21}, \dots, \alpha_{2s} | \dots | \alpha_{L1}, \dots, \alpha_{Ls}]^T \in \mathbb{R}$ is a set of coefficients.

In (8), most of the coefficients not associated with the class k will be close to 0. Hence α is a sparse vector, which can be calculated by solving the l_1 convex optimization problem or using iterative greedy pursuit algorithms such as orthogonal matching pursuit [23] and orthogonal least square [24]. Therefore, given an overcomplete dictionary $D \in \mathbb{R}^{m \times (s, L)}$ for L distinct classes and a test sample $y \in \mathbb{R}^m$, the SRC algorithm can be described by: (1) Solving the sparse representation $\hat{\alpha}$; (2) Computing the residual error e ; (3) Identifying y by $\arg \min_k e_k(y)$.

The basic idea of SRC is to find a sparse representation $\hat{\alpha}$ for an input test bio-code as a (sparse) linear combination of the overcomplete dictionary D , and then estimate the residual error e of each class iteratively by keeping the coefficients associated with that class and setting the remaining coefficients to 0. The class k associated with this test bio-code is then declared as the one that has the smallest approximation error. Similar to the use of majority voting for the reconciliation of noisy PUF response, noise in biometric feature extraction can also be mitigated by allowing a limited number of input image trials from the user. Sparsity Concentration Index (SCI) [25] can be used to assess the input image quality. If the image provided in the verification phase is analyzed to have an acceptable SCI, no further attempt is needed. Otherwise, another attempt can be made if the number of allowable attempts is not exceeded. If all the attempts are rejected, this particular (user, device) combination will be rejected directly.

D. UDhashing End-point Authentication Scheme

The parties involved are (at least) a user, (at least) a device, and a server. Our scheme enables the authentication between an endpoint UD and the server S . The trust model and the assumptions for the proposed system are given below:

Server S : The server S acts as a trusted party. It is assumed to have powerful resources (storage capacity and computational capability), and perform the enrollment of the device(s) and user(s) in a secure environment.

Endpoint UD : An endpoint is considered to be a (user, device) combination. The device is assumed to be equipped with an image sensor and a monoPUF. The device is deployed in an open and potentially harsh environment and may be compromised, lost or stolen.

Attacker E : E is assumed to have the capability to intercept the communication channel between S and UD . E can eavesdrop, manipulate, delete and reroute all protocol messages during transmission. Besides, E is assumed to be able to read any “secret information” stored in the non-volatile memory (NVM) of the endpoint device. However, E cannot access the challenge-response interface or he/she can at best read out a fraction of the responses of a strong PUF embedded in the endpoint device. E also cannot obtain temporary data (such as intermediate results of the computations) stored in the volatile memory of the device while the latter is participating in an authentication protocol as there are ways to design and implement the underlying algorithms such that their executions are side-channel aware [26].

An overview of the proposed system is shown in Fig. 1(c). The end-users are equipped with their end-devices, each

Fig. 2. UDhashing-based mutual authentication protocol. C_{uk}^i denotes the i^{th} C_u for the k^{th} user, $i \in \{1, 2, \dots, p\}$, similarly for R_{uk}^i ; C_j^k denotes the j^{th} C for the k^{th} user, $j \in \{1, 2, \dots, q\}$, similarly for R_j^k and $\tilde{T}_j^k$;

of which is embedded with a CMOS camera and a built-in monoPUF. The flow of the lightweight mutual authentication protocol depicted in Fig. 2 is detailed as follows:

Enrollment phase: The enrollment process is a one-time event that occurs in a secure environment at a trusted location. To initiate a legal enrollment process, the user needs to claim his/her eligibility to use the system. This can be accomplished by sending an “Enrol req” signal to the server containing the user’s verification information, which can be a verifiable ID such as national identity, passport, student matriculation, driving license, etc. issued by an agency that is accepted as legitimate, and can be used to match any additional personal information such as password or signature. The server will acknowledge the successful enrollment request by issuing the verified authorized user a user index. Upon receiving the “Enrol ack” signal from the server, the device index (i.e., MAC address) is sent to server as a record.

After a user has been authorized to register his/her device, the server issues a set of randomly generated challenge C_u and obtains its corresponding response set R_u from the device to be enrolled. The $C_u R_u$ pairs are stored in a secure server database $DB1$. Next, the user is required to take his/her face images, which will be stored temporarily in the device. Another set of randomly generated challenges $C \cap C_u = \emptyset$ is also sent to the endpoint device to generate the corresponding response R but R will not be sent to the server. Instead, a set of bio-codes T is generated from the enrolled images and R by our proposed UDhashing based scheme, and sent back to the server. The received bio-code for each (challenge, user, device) class are safely stored as a dictionary in $DB2$. There are several important points to note. Firstly, the interface for reading $C_u R_u$ pairs shall be disabled after enrollment, which can be achieved by an irreversible fuse or tamper-proof (public) storage of some sort [27]. Secondly, the stored face

images will be deleted immediately upon successful enrollment. Thirdly, the enrollment process is mostly conducted off line, with several face images to be taken from the user side. Fourthly, the bit length of PUF response for collecting $C_u R_u$ and CT pairs can be different, usually the latter is longer. This can be accomplished by a series of internal challenges generated from C through a LFSR. A longer PUF response R can be obtained by concatenating the responses to multiple internal challenges for the generation of the bio-code T . Lastly, the cardinalities of C_u and C are p and q respectively, which are determined based on the use case.

Mutual authentication phase:

- A user can seek access permission to the system with his/her device. To prevent any attack from illegally obtaining the bio-codes, the server must be authenticated by the endpoint first.
- After receiving the access request by claiming the user index U_k and device index D_k , a challenge C_{uk}^i , $i \in \{1, 2, \dots, p\}$, from the enrolled set C_{uk} for the k^{th} user is randomly selected and sent to the endpoint. At the endpoint, a random number N is generated by a low-power TRNG such as [28], which can prevent the attacker from impersonating a server by repeatedly collecting the same authentication credentials to exploit PUF noise as side-channel information. A lightweight hash function such as Spongent [29] is applied to N and R_{uk}^i to generate a hash value A before N is sent to the server.
- The server retrieves R_{uk}^i from his database, and calculates B using the same hash function after receiving N . B is then sent back to the endpoint.
- The endpoint compares the received B with A . If $A \neq B$, the communication is aborted. If $A = B$, an acknowledge signal ACK will be sent back to the server. An attacker who has no knowledge about the enrolled CRPs of the embedded PUF cannot impersonate the server as he cannot find a correct R_{uk}^i to make $A = B$. The device-side random nonce N prevents the attacker from using any eavesdropped (C_{uk}^i, B) pairs to gain authentication.
- The server then issues a challenge C_j^k , $j \in \{1, 2, \dots, q\}$ randomly selected from the enrolled set C^k to the endpoint. Upon receiving C_j^k , the endpoint generates a “live bio-code” $\tilde{T}_j^k$ according to the proposed UDhashing scheme, and this bio-code is sent back to the server for authentication.
- The received “live bio-code” is verified using the SRC method. If the computed class y matches the claimed class k with the least approximation error, the authentication is successful. Otherwise the request will be rejected.

The CRP database ($DB1$ in Fig. 2) can also be replaced by a PUF model with a strong PUF that supports model-based authentication [27], [30]. Utilizing strong PUF model increases the protocol flexibility and reduces the database size. The accuracy of the PUF model is important to ensure the correct generation of R_{uk}^i at the server in Fig. 2. In addition, the above authentication process can be conducted multiple times by sending different enrolled challenges to the device to reduce the false negatives caused by imperfect regeneration of face images, unstable PUF response and transmission error. In our proposed scheme, once the bio-code for a (challenge,

user, device) combination transmitted over the channel is compromised, the corresponding tuple will be deleted from the system. The system can be updated with new bio-codes easily by applying different challenges to the endpoint during the enrollment phase. Besides, since we use SRC for recognition, adding new combinations to the existing database is straightforward. To add new users, all that is needed is to obtain their bio-codes, and add them into the secure database. In our previously reported scheme [10], adding new users will require all the existing accounts to be re-enrolled, which is impractical and infeasible in real applications.

IV. RESULTS AND DISCUSSIONS

A. Dataset and Experimental Setting

The experimental results are evaluated using the ORL [31] and Ext. Yale B [32] face recognition databases. ORL is used as the genuine users while Ext. Yale B is used as the impostors. ORL database contains 40 distinct subjects, each has 10 different images taken at different time with varying illumination and facial expressions or details. The first 6 images of each user in the ORL database are selected for authorized user enrollment, while the remaining 4 images are used in the authentication phase for system performance test. Ext. Yale B contains 38 subjects of 9 poses and 64 illumination conditions. For our experiments, the images with the most neutral light sources (“A+000E+00”) were used as the impostor database. As for the device key extractor, the measured silicon data from 20 monoPUF chips [22] were used. Their challenge bit length is 64 bits. Among these 20 chips, 10 are used as the trusted devices, while the remaining 10 are used as the malicious devices. Altogether there are 38 impostors and 10 untrusted devices that were not enrolled in the system. Due to the limited number of fabricated dies, these devices are shared among different users to form different (user, device) pairs.

Each raw image is pre-processed (e.g. size normalization) into 128×128 gray image. 2560 Gabor features are obtained by applying a set of Gabor filters (5 scales and 8 orientations) with a down-sampling rate of 256. Besides, 1600 DCT-II features are obtained by first dividing the image into 16 blocks of size 32×32 each, and then extracting the first 100 DCT coefficients. By concatenating these two features, a 4160-dim facial biometric feature is obtained. The concatenated features are divided into 13 groups each of 320 bits, so that each subgroup can have the same size of maximum-bit unitary features. These feature clips are used as the inputs of the UDhashing scheme.

B. Selection of Optimal Bio-code Dimension

As indicated by the John-Lindenstrauss Lemma, for a given data set, a larger bio-code dimension m means a smaller ϵ , which implies a better preservation of the pairwise distance between any two data points after projection. However, increasing m requires PUF-based device key to be proportionally expanded, and takes longer time to run the mGS orthogonalization. To find a good cost-performance trade-off in the selection of bio-code dimension m , we gradually increase

Fig. 3. Distribution of v and its corresponding mean μ for each m .

the sub-group feature dimension m' ($m = 13 \times m'$) from 5, 10, 50, 100, 200 to 320 (the original sub-feature dimension). All these matrices are verified to have full column rank of m' . For each case of m , the quantization threshold τ is first determined according to the distribution of v in (2) before the approximation of $RR^T \approx I$ and the resultant authentication performance are evaluated.

Fig. 3 shows that v for each $m'(m)$ is approximately normally distributed. The perturbation due to the variation of m' is insignificant and the means of the distributions for all m' remain infinitesimally close to zero. Therefore, the quantization threshold $\tau = 0$ is applicable for all cases. As analyzed in Sec. II-A, $RR^T \approx I$ only if $\delta_{i,j} = RR^T - I \approx 0$. The measured mean μ_δ and standard deviation σ_δ of $\delta_{i,j}$ shown in Table I decreases with increasing m . For semi-orthogonal matrix R , a small number of outliers are found in the distribution of $\delta_{i,j}$, but the magnitudes of outliers decreases with increasing m . Hence, it can be concluded that increasing m improves the approximation of $RR^T \approx I$.

TABLE I
MEAN δ_μ AND STANDARD DEVIATION δ_σ OF $\delta_{i,j}$ WITH VARYING m .

$m'(m)$	outliers	μ_δ	σ_δ
5(65)	$[-1, -0.402]$	-5.1×10^{-4}	0.0555
10(130)	$[-1, -0.402]$	-2.8×10^{-4}	0.0550
50(650)	$[-0.89, -0.791]$	-5.4×10^{-5}	0.0513
100(1300)	$[-0.749, -0.45]$	-2.0×10^{-5}	0.0464
200(2600)	$[-0.487, -0.188]$	-6.0×10^{-6}	0.0342
320(4160)	—	3.2×10^{-19}	3.92×10^{-15}

Fig. 4 shows the normalized Hamming distance (HD) distributions of the (genuine user, genuine device), (fake user, genuine device) and (genuine user, fake device) combinations. It shows that increasing m' will shrink the distribution curve, and result in a lower false acceptance rate or false rejection rate. The standard deviation converges as m' increases beyond 50. The corresponding authentication performances are shown in Table II. The final authentication decision of accept/reject is determined using the SRC algorithm rather than simply comparing the normalized HD against a threshold. Therefore even if overlaps may exist between the distribution curves of genuine and malicious cases in Fig. 4, they may still be correctly classified due to the robustness of SRC. Based on the experimental results, when $m' = 100$ ($m = 1300$), perfect authentication can be achieved for all the combinations.

TABLE II
AUTHENTICATION PERFORMANCE WITH VARYING m .

$m'(m)$	5(65)	10(130)	50(650)	100(1300)	200(2600)	320(4160)
genuine	0.8125	0.9437	0.9875	1	1	1
fake user	0.0395	0.0158	0.0053	0	0	0
fake device	0.0063	0	0	0	0	0

Therefore, the dimension of each sub-group bio-code is set to be 100, giving a total bio-code length of 1300. To produce a (100×13) -bit transformed bio-code, responses of length 320×100 bits are collected from each PUF.

C. UDhashing Discriminability

The discriminability of UDhashing describes its capability to detect changes made to any of the three entries of a (challenge, user, device) tuple. A good UDhashing should be sensitive to variability in challenge, device and user. Table III shows all valid use cases. For an effective endpoint authentication system, when given a specific challenge, only the authorized user with trusted devices (genuine case) can be authenticated successfully, other cases like malicious user (case 3), unregistered devices (case 4) or their combination (case 5) should be rejected. The bio-code should also be updatable (case 2).

TABLE III
UDHASHING DISCRIMINABILITY TEST.

Cases	Genuine case	Case 2	Case 3	Case 4	Case 5
Challenge	same	different	same	same	same
User	same	same	different	same	different
Device	same	same	same	different	different

The UDhashing discriminability can be evaluated by the normalized Hamming distance (HD) between the obtained “live bio-codes” and the enrolled bio-codes. Let N_{usr} denote the number of users enrolled in the system, and N_t the number of trials per user in the authentication phase, then the discriminability D can be calculated as follows:

$$D = \frac{1}{N_{usr} \times N_t} \sum_{i=1}^{N_{usr}} \sum_{j=1}^{N_t} \frac{HD(T_i, T_j)}{l} \quad (9)$$

where $HD(T_i, T_j)$ is the HD between two binary bio-code strings T_i and T_j , and l is the bit length of a string.

Fig. 5 shows the normalized HD distribution for the five cases listed in Table III. For a good discriminability, the mean (μ) normalized HD for each impostor combination should be 0.5, while the ideal standard deviation (σ) for a secure m -bit

Fig. 5. Discriminability against: (a) Genuine case: $\mu = 0.265$, $\sigma = 0.065$; (b) Case 2: $\mu = 0.499$, $\sigma = 0.013$; (c) Case 3: $\mu = 0.497$, $\sigma = 0.015$; (d) Case 4: $\mu = 0.502$, $\sigma = 0.014$; (e) Case 5: $\mu = 0.497$, $\sigma = 0.015$.

bio-codes is $\sqrt{m}/(2m)$ (ideal $\sigma = 0.014$ for $m = 1300$) [33]. The distribution of all the impostor cases are clustered around 0.5 with a standard deviation close to the ideal value while the mean of the genuine case is measured to be 0.266. The results show that the proposed scheme has good discriminability.

D. Authentication Performance

Authentication performance is determined by the accuracy of matching the actual identity to the claimed identity. The genuine case should have high similarity score (ideally 100%) while the impostor cases should have low score (ideally 0%). Besides, the performance of the original system components should not be undermined by the implementation of UDhashing. In this experiment, the accuracy of both normal face authentication system (without UDhashing) and the proposed system (with UDhashing) were tested. Table IV shows that all five cases are correctly detected by our proposed system. The performance has even exceeded the normal authentication system for the two applicable use cases owing to the increase in inter-class distances by the (user, device)-specific random projections.

TABLE IV
VERIFICATION PERFORMANCE

Accuracy \ Algorithm	Without UDhashing	With UDhashing
Genuine case	0.9812	1.0
Case 2	N.A	0.0
Case 3	0.0	0.0
Case 4	N.A	0.0
Case 5	N.A	0.0

Fig. 4. Normalized Hamming distance distribution for (a) (genuine user, genuine device) (b) (fake user, genuine device) (c) (genuine user, fake device) combinations with varying m .

E. Device Key Reliability Issues

The random matrix, and hence the projection space for facial biometrics, may be affected by the reproducibility of PUF response to the same challenge caused by the change in temperature and supply voltage. To determine the inherent error tolerance of UDhashing against PUF response reliability, we randomly flip certain percentage of bits in the genuine PUF response matrix, and calculate the corresponding authentication accuracy. Fig. 6(a) shows that the authentication accuracy can still be maintained at an acceptably high level (> 0.95) by keeping the BER of PUF response below 15%. This has been

well satisfied by the proposed monoPUF [22] for temperature varies from $-40^\circ \sim 90^\circ$ and supply voltage ranges from $0.9 \sim 1.3V$.

Fig. 6. (a) Authentication accuracy with unreliable device keys; (b) FAR versus average bit error rate of genuine bio-codes.

F. Comparison with Existing Works

Authentication schemes that unify both user and device characteristics are rare. Table V summarizes the salient features of existing user-device bipartite authentication schemes found in the literature. The proposed method is more versatile in terms of system reconfigurability, cancelable biometrics, mutual authenticity and performance, making it a strong complement to the match-on-device authentication scheme of [9].

Compared to the classical biometric authentication system, the cryptographic primitives used by the endpoint device are only a monoPUF [22], a TRNG [28] and a Spongent hash [29]. The 64-bit monoPUF consumes merely $3.85 \mu W$ power with an active silicon area of only $4719 nm^2$ [22]. The reported ASIC implementation of Spongent-128 consumes only 1060 gate equivalents and $2.2 \mu W$ [29], and TRNG consumes 214 nW at a throughput of 500 kbps with an area of $1400 \mu m^2$ [28]. Since these components are on standby until triggered by the authentication request, their actual power consumptions are lower. Moreover, our proposed scheme does not require compute-intensive encryption algorithms like AES and RSA, which are used to protect the biometric templates in typical biometric authentication system [5]. The signal processing cost in UDhashing has also been largely reduced by the dimension reduction of raw biometric features.

V. SECURITY ANALYSIS

A. Memory Probing

UDhashing achieves cancelable biometrics as well as device tracking by PUF, which offers tamper-evident and secret-destruction features against physical probing. At the endpoint, the face images are captured in real-time while the PUF response is only generated upon request. The cancelable bio-codes that indistinguishably unifying the biometric and device information are sent directly to the verifier for later classification. The entire process requires no secret data to be stored in the endpoint device. Therefore, our UDhashing scheme does not suffer from the same security threats as traditional biohashing schemes which have their random number or TPRN seed stored in the NVM of the device for random projection when the device is lost, stolen or compromised.

B. Random Guessing

The most common case is the attacker trying to access the system by randomly guessing the authentication credential, i.e., the bio-code, without the authentic user's biometric and the correct device. The attacker may randomly send an authentication ID to request for authentication. Upon receiving the challenge from the server, he/she may determine the hashed bio-code by trial and error. Since the system identifies a requester according to the minimum residual error, he/she does not need to correctly guess all bits of a valid bio-code. To evaluate the probability of a successful attack, we randomly flipped certain percentage of bits in each bio-code of a genuine case to determine the corresponding false accept rate (FAR). Fig. 6(b) shows that the FAR is low (≈ 0.0625) with 40% of errors (or the fraction of correctly guessed bits $f = 0.6$) in the guessed bio-codes. The FAR reduced to 0 when the $f \leq 1 - 0.45 = 0.55$. Assume that bits 0 and 1 occur with equal probability in each valid bio-code, the probability of correctly guessing at least a fraction f of an N -bit bio-code is given by

$$P = \sum_{i=\lceil Nf \rceil}^N \binom{N}{i} \left(\frac{1}{2}\right)^i \left(1 - \frac{1}{2}\right)^{N-i} = 2^{-N} \sum_{i=\lceil Nf \rceil}^N \binom{N}{i} \quad (10)$$

To achieve a 6.25% FAR, the attacker needs to guess at least 60% of bits of a valid bio-code correctly. For $N = 1300$, $f = 0.6$, P is around 2.86×10^{-13} . Hence, even just $\approx 6.25\%$ of false acceptance is close to impossible.

Another situation is that the attacker may have obtained some noisy biometric such as synthetic samples or photographs of an authentic user. Assume that the noisy biometric is accurate enough to represent the authentic user but the attacker does not have the correct device. In this case, the attacker may try to generate an effective bio-code by randomly guessing the correct PUF response. The attacker may exploit the system tolerance to some bit errors in the PUF response as analyzed in Sec. IV-E. To obtain a minimum above zero authentication accuracy of 3.13% based on the results of Fig. 6(a), the attacker needs to guess at least 60% of PUF response bits correctly. According to (10), for $N = 32000$, $f = 0.6$, P is less than 2.2×10^{-282} , which is close to impossibility. To achieve larger authentication accuracy, the attacker will have to guess more PUF response bits correctly, and the probability P will be even smaller.

C. Replay Attack

In the proposed system, the hashed bio-code is transferred from an end-point device to a verifier, and serves as a direct credential for classification. Replay attack could be conducted by an attacker by eavesdropping the login credentials (the bio-codes), and then waiting for the same challenge from the verifier to replay the hijacked bio-code. The attacker may collect many challenge bio-code pairs (CBPs) of different genuine user-device bio-codes to increase his chance of success. However, as every unique challenge is used only once, all the used CBPs collected by the attacker are useless for future replay. The use of strong PUF ensures that enough CBPs can

TABLE V
COMPARISON OF USER-DEVICE BIPARTITE AUTHENTICATION SCHEMES.

Scheme	Matching Site	System reconfigurability*	Cancelable biometrics	Mutual authenticity	Performance**
R.A.Scheel2015 [7]	Not specified	No	No	No	Moderate***
Y.Guo2017 [8]	Not specified	No	No	No	Moderate
N.Karimian2018 [9]	Match-on-device	No	Yes	No	Good
Y.Zheng2018 [10]	Match-on-server	Yes	Yes	No	Good
This work	Match-on-server	Yes	Yes	Yes	Good

* Ability to re-issue an authentication credential/key for the same (user, device) combination. ** Reproducibility for genuine case and discriminability against malicious cases.

*** Inadequate sample size and inconclusive discriminability test.

be generated for authentication throughout the lifetime of the device.

D. Man-in-the-middle Attack

MITM is an attack whereby an attacker intercepts the communication channel to actively eavesdrop, replay or secretly alter the information transmitted between two parties. The attacker who plays MITM may masquerade as a legitimate endpoint, or impersonate the legitimate server to collect the login credentials. In our proposed scheme, an attacker who does not possess a registered device or privilege to access the secure remote CRP database is unable to generate the correct response to successfully masquerade as the registered device or impersonate the server. It is also impossible for the attacker to collect or deduce the CRPs as the raw response generated by the PUF is never directly transmitted over the channel during protocol execution. Therefore, the attacker can neither calculate the correct B to impersonate a server nor provide a correct $\tilde{T}_j^k$ to masquerade as an authorized endpoint.

E. Modeling Attack

The attacker may machine learn the CRPs of the embedded strong PUF of a registered device to correctly predict the responses to future challenges. To train the machine model, the attacker will have to collect a sufficient number of CRPs from a registered device. Ultimately, the PUF and UDhashing modules are monolithically integrated and only the “live bio-code” $\tilde{T}_j^k$ is directly accessible from the outputs of the module if the attacker has a temporary possession of a registered device. To probe into the module internals for CRP collection, the attacker needs longer possession of the device, which is possible with a lost or stolen device. As the response can only be generated with the circuit turned on, invasive and semi-invasive attacks can easily alter the subtle parameters during operation, resulting in invalid measured CRPs. Even if the attacker can successfully model the PUF through physical attack on a lost or stolen registered device, due to the cancelable biometrics, the bio-code generated using the correctly predicted response is unlikely to be accepted if the owner has reported the lost device and refreshed his/her bio-code with a new device. To be on the safe side, modeling-attack resilient strong PUF such as monoPUF is adopted in the proposed system. It was demonstrated that modeling accuracy of advanced support vector machines [34] and reliability-based covariance matrix adaptation evolution strategy [21], even with a large training set of ten thousand CRPs, on monoPUF has been reduced to around 50% [22]. If physical attack on PUF fails, black-box modeling of the Biohashing module by using

CBPs for training will be even less likely to succeed due to the additional highly random and nonlinear mixing of different distributions of physical and non-physical random variables.

VI. CONCLUSION

In this paper, a novel UDhashing based mutual authentication scheme is proposed. Its security over our preliminary proposal [10] has been enhanced by utilizing a modeling attack resilient strong PUF and optimizing the authentication protocol to minimize privacy leakage and conserve transmission bandwidth. Facial biometrics and PUF are integrated efficiently by UDhashing to provide cancelable biometrics along with bipartite authentication of both user and device. The system can update the dictionary for renewal of bio-codes by issuing different challenge sets to the endpoints. A strong PUF, a TRNG and a lightweight hash are the only cryptographic primitives needed in the endpoint-device side for the application of the proposed scheme. Our experimental results corroborate that UDhashing has good authentication accuracy, and can well distinguish between genuine and impostor (challenge, user, device) tuples. It avoids the weak assumption of conventional biohashing schemes, and is analyzed to be resilient against random guessing, replay, MITM and modeling attacks. Other biometric features like iris, ECG, etc. can also be adapted for the projection into the PUF-defined device-specified random space. The obtained unified authentication credential can be input to other biometric recognition algorithm besides SRC.

REFERENCES

- [1] VMware, “End-user device registration,” 2018. [Online]. Available: <https://docs.vmware.com/en/VMware-AirWatch/9.2/aw-mdm-guide-92/GUID-014MDM-EndUserDeviceRegistration.html>
- [2] Microsoft, “Link users and devices with user device affinity in system center configuration manager,” 2016. [Online]. Available: <https://docs.microsoft.com/en-us/sccm/apps/deploy-use/link-users-and-devices-with-user-device-affinity>
- [3] SolarWinds, “User device tracker.” [Online]. Available: <https://www.solarwinds.com/user-device-tracker>.
- [4] 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, DOI 10.1109/MCAS.2017.2713305, no. 3, pp. 32–62, 2017.
- [5] K. Nandakumar and A. K. Jain, “Biometric template protection: Bridging the performance gap between theory and practice,” *IEEE Signal Process. Mag.*, vol. 32, DOI 10.1109/MSP.2015.2427849, no. 5, pp. 88–100, Aug. 2015.
- [6] L. Zhang, Y. Zhang, S. Tang, and H. Luo, “Privacy protection for e-health systems by means of dynamic authentication and three-factor key agreement,” *IEEE Trans. Ind. Electron.*, vol. 65, DOI 10.1109/TIE.2018.2823686, no. 3, pp. 2795–2805, Apr. 2018.
- [7] R. A. Scheel and A. Tyagi, “Characterizing composite user-device touchscreen physical unclonable functions (PUFs) for mobile device authentication,” in *Proc. 5th Int. Workshop Trustworthy Embedded Devices*, DOI 10.1145/2808414.2808418, pp. 3–13, Denver, Colorado, USA, Oct. 2015.

- [8] Y. Guo and A. Tyagi, "Voice-based user-device physical unclonable functions for mobile device authentication," *J. Hardware Syst. Security*, vol. 1, DOI 10.1007/s41635-017-0003-4, no. 1, pp. 18–37, May. 2017.
- [9] K. et al., "Secure and reliable biometric access control for resource-constrained systems and IoT," *arXiv preprint arXiv:1803.09710*, 2018.
- [10] Y. Zheng, Y. Cao, and C.-H. Chang, "Facial bihashing based user-device physical unclonable function for bring your own device security," in *Proc. IEEE Int. Conf. Consumer Electronics*, DOI 10.1109/ICCE.2018.8326074, pp. 1–6, Las Vegas, USA, Mar. 2018.
- [11] S. Dasgupta and A. Gupta, "An elementary proof of the Johnson-Lindenstrauss lemma," *Int. Computer Science Institute, Technical Report*, vol. 22, no. 1, pp. 1–5, 1999.
- [12] S. S. Vempala, *The Random Projection Method*, vol. DIMACS-65. American Mathematical Soc., 2005.
- [13] A. B. Teoh, Y. W. Kuan, and S. Lee, "Cancellable biometrics and annotations on bihash," *Pattern Recognition*, vol. 41, DOI 10.1016/j.patcog.2007.12.002, no. 6, pp. 2034–2044, Jun. 2008.
- [14] J. K. Pillai, V. M. Patel, R. Chellappa, and N. K. Ratha, "Secure and robust iris recognition using random projections and sparse representations," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 33, DOI 10.1109/TPAMI.2011.34, no. 9, pp. 1877–1893, Sept. 2011.
- [15] Y. Liu and D. Hatzinakos, "Biohashing for human acoustic signature based on random projection," *Canadian J. Electr. Computer Eng.*, vol. 38, DOI 10.1109/CJECE.2015.2416200, no. 3, pp. 266–273, Summer 2015.
- [16] L. Giraud, J. Langou, M. Rozložník, and J. van den Eshof, "Rounding error analysis of the classical Gram-Schmidt orthogonalization process," *Numerische Mathematik*, vol. 101, DOI 10.1007/s00211-005-0615-4, no. 1, pp. 87–100, Jul. 2005.
- [17] Å. Björck and C. C. Paige, "Loss and recapture of orthogonality in the modified Gram-Schmidt algorithm," *SIAM J. Matrix anal. Appl.*, vol. 13, DOI 10.1137/0613015, no. 1, pp. 176–190, 1992.
- [18] A. Kong, K.-H. Cheung, D. Zhang, M. Kamel, and J. You, "An analysis of bihashing and its variants," *Pattern Recognition*, vol. 39, DOI 10.1016/j.patcog.2005.10.025, no. 7, pp. 1359–1368, Dec. 2005.
- [19] V. Štruc and N. Pavešić, "The complete gabor-fisher classifier for robust face recognition," *EURASIP J. Advances in Signal Process.*, vol. 2010, DOI 10.1155/2010/847680, p. 31, May. 2010.
- [20] Y. Zheng, S. S. Dhabu, and C.-H. Chang, "Securing IoT monitoring device using PUF and physical layer authentication," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, DOI 10.1109/ISCAS.2018.8351844, Florence, May. 2018.
- [21] G. T. Becker, "On the pitfalls of using arbiter-PUFs as building blocks," *IEEE Trans. Computer-Aided Design Integr. Circuits and Syst.*, vol. 34, DOI 10.1109/TCAD.2015.2427259, no. 8, pp. 1295–1307, Apr. 2015.
- [22] Y. Cao, C. Q. Liu, and C.-H. Chang, "A low power diode-clamped inverter based strong physical unclonable function for secure and robust IoT authentication," *IEEE Trans. Circuits Syst.-I: Reg. Papers*, DOI 10.1109/TCSI.2018.2855061, no. 99, pp. 1–10, Aug. 2018.
- [23] B. Mamandipoor, D. Ramasamy, and U. Madhow, "Newtonized orthogonal matching pursuit: Frequency estimation over the continuum," *IEEE Trans. Signal Process.*, vol. 64, DOI 10.1109/TSP.2016.2580523, no. 19, pp. 5066–5081, Oct. 2016.
- [24] M. Cui and S. Prasad, "Sparse representation-based classification: Orthogonal least squares or orthogonal matching pursuit?" *Pattern Recognition Lett.*, vol. 84, DOI 10.1016/j.patrec.2016.08.017, pp. 120–126, Aug. 2016.
- [25] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, DOI 10.1109/TPAMI.2008.79, no. 2, pp. 210–227, Feb. 2009.
- [26] A. Van Herrewewe *et al.*, "Reverse fuzzy extractors: Enabling lightweight mutual authentication for PUF-enabled RFIDs," in *Proc. Int. Conf. Financial Cryptography Data Security*, pp. 374–389. Springer, Feb.-Mar. 2012.
- [27] M. Yu *et al.*, "A lockdown technique to prevent machine learning on PUFs for lightweight authentication," *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 2, DOI 10.1109/TMSCS.2016.2553027, no. 3, pp. 146–159, Jul. 2016.
- [28] T.-K. Kuan, Y.-H. Chiang, and S.-I. Liu, "A 0.43 pJ/bit true random number generator," in *Proc. Asian Solid-State Circuits Conf.(A-SSCC)*, DOI 10.1109/ASSCC.2014.7008853, pp. 33–36. KaoHsiung, Taiwan: IEEE, Nov. 2014.
- [29] A. Bogdanov *et al.*, "SPONGENT: A lightweight hash function," in *Proc. Int. Workshop Cryptographic Hardware. Embedded Syst.*, pp. 312–325. Springer, 2011.
- [30] S. S. Zalivaka, A. A. Ivaniuk, and C.-H. Chang, "Reliable and modeling attack resistant authentication of arbiter PUF in FPGA implementation with trinary quadruple response," *IEEE Trans. Inf. Forensics and Security (TIFS)*, DOI 10.1109/TIFS.2018.2870835, pp. 1–1, Sept. 2018.
- [31] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Proc. 2nd IEEE Workshop. Appl. Computer Vision*, pp. 138–142, Sarasota, USA, Dec. 1994.
- [32] Yale, "The extended yale face database B," 2001. [Online]. Available: <http://vision.ucsd.edu/~iskwak/ExtYaleDatabase/ExtYaleB.html>
- [33] K. Fruhashi, M. Shiozaki, A. Fukushima, T. Murayama, and T. Fujino, "The arbiter-PUF with high uniqueness utilizing novel arbiter circuit with delay-time measurement," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, DOI 10.1109/ISCAS.2011.5938068, pp. 2325–2328, Rio, Brazil, May. 2011.
- [34] U. Rührmair *et al.*, "PUF modeling attacks on simulated and silicon data," *IEEE Trans. Inf. Forensics Security*, vol. 8, DOI 10.1109/TIFS.2013.2279798, no. 11, pp. 1876–1891, Nov. 2013.

Yue Zheng (S'2015) received the B. Eng. Degree in the School of Communication and Information Engineering from Shanghai University (SHU), China in 2015. She is currently working towards the Ph.D. degree in the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU) Singapore. Her areas of research interest are physical unclonable functions, device fingerprinting, and hardware security.

Yuan Cao 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 an assistant 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 four books, published ten book chapters, around 100 international journal papers (two-thirds are IEEE) and more than 170 refereed international conference papers (mostly in IEEE), and delivered over 30 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.