Physical unclonable function based solutions to unification of user, device and data authentication

Zheng, Yue

2020

Zheng, Y. (2020). Physical unclonable function based solutions to unification of user, device and data authentication. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/137808>

<https://doi.org/10.32657/10356/137808>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 19:16:27 SGT

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

**PHYSICAL UNCLONABLE FUNCTION BASED
SOLUTIONS TO UNIFICATION OF USER,
DEVICE AND DATA AUTHENTICATION**

ZHENG YUE

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University in partial
fulfillment of the requirements for the degree of
Doctor of Philosophy

2020

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

10/Apr/2020

Date

ZHENG YUE

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

10/Apr/2020

Date

CHANG Chip Hong

Authorship Attribution Statement

This thesis contains materials from four papers that have been published in the following peer-reviewed journals where I am the first student author.

The content of Chapter 2 has been published as 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, pp. 32–62, thirdquarter 2017.

The contributions of the co-authors are as follows:

- A/Prof. Chang Chip Hong prepared the proposal of this paper, provided the guidance on the paper organization, analyzed the research status and future directions and edited the manuscript drafts.
- I performed all the literature review, made initial manuscript draft of surveyed topics and provided additional inputs to the content description for A/Prof. Chang’s edition.
- Dr. Zhang Le assisted in manuscript revision.

The content of Chapter 3 has been published as Y. Zheng, Y. Cao, and C. H. Chang, “UDhashing: Physical unclonable function based user-device hash for endpoint authentication,” *IEEE Trans. Industrial Electronics (TIE)*, vol. 66, pp. 9559–9570, Jan. 2019.

The contributions of the co-authors are as follows:

- A/Prof. Chang Chip Hong initiated the concept, provided the initial project direction, guided the system analysis and edited the manuscript drafts.
- I performed all the literature review, designed the UDhashing scheme and evaluation methods. The experiments were conducted and analyzed by me and I also drafted the first manuscript.
- Dr. Cao Yuan helped to revise the manuscript and gave suggestions on the interpretation of experimental data.

The content of Chapter 4 has been published as Y. Zheng, Y. Cao, and C. H. Chang, “A PUF-based data-device hash for tampered image detection and source camera identification,” *IEEE Trans. Inf. Forensics, Secur. (TIFS)*, vol. 15, pp. 620–634, Jul. 2019.

The contributions of the co-authors are as follows:

- A/Prof. Chang Chip Hong initiated the idea, provided the initial project direction and guidance in use-case definition as well as the system analysis. A/Prof Chang Chip Hong also edited the manuscript drafts.
- I performed all the literature review, designed the perceptual data-device hash system and the evaluation methods. I also conducted all the experiments and analyses, and prepared the manuscript drafts.
- Dr. Cao Yuan helped to revise this manuscript and gave suggestions to improve my initially proposed PUF-based Knuth Shuffle algorithm design.

The content of Chapter 5 has been published as Y. Zheng, X. Zhao, S. Takashi, Y. Cao, and C. H. Chang, “Ed-PUF: Event-driven physical unclonable function for camera authentication in reactive monitoring system,” *IEEE Trans. Inf. Forensics, Secur. (TIFS)*, pp. 1–1, Mar. 2020.

The contributions of the co-authors are as follows:

- A/Prof. Chang Chip Hong introduced the notion of event-driven PUF, provided the initial project direction, and gave advices on hardware lock design as well as security analyses. A/Prof. Chang Chip Hong also edited the manuscript drafts.
- I performed all the literature review, introduced and designed the event-driven PUF system and the evaluation methods. I also conducted all the experiments and analyses, and prepared the manuscript drafts.
- Dr. Zhao Xiaojin helped with the simulation of the raw event-driven PUF in Cadence and assisted in the revision of the manuscript.
- Dr. Takashi Sato provided suggestions on secure transmission scheme and experiment design. He also helped to revise the manuscript.
- Dr. Cao Yuan helped to revise this manuscript and provided valuable advices on the DVS image pixel design and asynchronous event read out operations.

10/Apr/2020

Date

ZHENG YUE

Acknowledgements

I would like to express my deepest gratitude to my supervisor: Associate Professor Chang Chip Hong, for his constant support, heuristic teaching and kind patience. Prof. Chang is always passionate and critical about research, which has greatly inspired me during my Ph.D. candidature. His acute sense of finding good research problems, rigorous scholarship and commitment to research integrity have set a good example for us in the way of being a qualified research scientist. Besides, I would also like to thank Mrs Chang, who together with Prof. Chang, treats us like family members. I will never forget those bowling sessions, new year gatherings, birthday wishes, travelling times after conferences...which give me great warmth and courage to live and study in a new environment that is far away from my home country.

I would also like to express my appreciation to those who helped me during the four years of research journey. Thanks to Professor Takashi Sato for providing me the opportunity of an exchange program in Kyoto University and the valuable suggestions given to my research works. Thanks to Professor Zhao Xiaojin for his generous support on tape out opportunities and guidances as well as those great discussions on our collaborative projects. Thanks to my seniors, Dr. Cao Yuan, Dr. Sumedh Somnath Dhabu, Dr. Zhang Le, Dr. Sachin Kumar, Dr. Siarhei Zalivaka from our research group, and Dr. Yang Yongkui, Dr. Zhang Xiangyu, Dr. Jiang jize from other research groups, etc., for giving me great technical help and guidance in doing projects. Thank all my friends, Dr. Ho Truong Phu Truan, Mr. Liu Chaoqun, Ms. Wang Si, Dr. Zhang Ruochong, Dr. Liu Siyu, Mr. Yang Chuanshi, Dr. Jin Haoran, Ms. Luo yunqi, Ms. Liu Shuangli, etc., for giving me great accompany and support. Besides, I would also like to thank VIRTUS technicians, Ms. Gek Eng Tan and Ms. Guee Geok Lian for their assistance.

Finally, I would like to express my special thanks to my beloved ones. My parents, Zheng Chengde and Zou Honghua, always give me endless love and unconditional support in the journey of pursuing higher degree. My honor belongs to you. I love you forever and always wish you good health and happiness. My little sister, Zheng Zhao, always prepares gifts before I go home and cries each time when I have to leave. I wish you success in the coming Chabansheng exams and have a wonderful and fulfilling college life ahead. Also, thanks to my boyfriend, Liu Wenye, who loves me, respects me and cares for me, I wish you great success in your Ph.D. and I will always be a strong supporter in your life.

Table of contents

List of figures	xv
List of tables	xix
Summary	xxii
1 Introduction	1
1.1 Motivation	1
1.2 Research Objectives	5
1.3 Major Contributions	7
1.3.1 A PUF-based User-Device Hash for Endpoint Authentication	7
1.3.2 A PUF-based Data-Device Hash for Digital Forensics	8
1.3.3 An Event-Driven PUF for Reactive Monitoring System	8
1.4 Organization	9
2 Literature Review	11
2.1 PUF Preliminaries	11
2.1.1 Definition	11
2.1.2 Evaluation Metrics	11
2.2 PUF Categories	15
2.2.1 Weak PUF	15
2.2.2 Strong PUF	18
2.3 Related Works	22
2.3.1 Sensor-based PUF	22
2.3.2 User-Device Authentication	25
2.3.3 Data-Device Authentication for Digital Forensics	27
2.4 Summary	28

Table of contents

3	UDhashing: Physical Unclonable Function Based User-Device Hash for Endpoint Authentication	31
3.1	Related Works	33
3.1.1	Cancelable Biometrics based on Random Projection	33
3.1.2	Constraints of Existing Biohashing Schemes	35
3.2	Proposed Cancelable UDhashing Scheme	35
3.2.1	Facial Feature Representation	35
3.2.2	Device Key Generation	37
3.2.3	Bio-code Generation and Recognition	38
3.2.4	UDhashing End-point Authentication Scheme	40
3.3	Results and Discussions	44
3.3.1	Dataset and Experimental Setting	44
3.3.2	Selection of Optimal Bio-code Dimension	44
3.3.3	UDhashing Discriminability	46
3.3.4	Authentication Performance	48
3.3.5	Device Key Reliability Issues	48
3.3.6	Comparison with Existing Work	49
3.4	Security Analysis	50
3.4.1	Memory Probing	50
3.4.2	Random Guessing	50
3.4.3	Replay Attack	51
3.4.4	Man-in-the-middle Attack	52
3.4.5	Modelling Attack	52
3.5	Summary	53
4	A PUF-based Data-Device Hash for Tampered Image Detection and Source Camera Identification	55
4.1	Preliminaries	57
4.1.1	Speeded up robust features	57
4.1.2	M-estimator Sample Consensus	58
4.1.3	Physical Unclonable Function	59
4.2	Proposed PUF-based Data-device Hash	60
4.2.1	Robust Feature Extraction	60
4.2.2	Perceptual Data-Device Hash Generation	62
4.2.3	Image Tampering Detection and Source identification	64
4.3	Experiment Setting and Key Parameter Optimization	68
4.3.1	Dataset Preparation	68

4.3.2	Hash Dimension Selection	70
4.3.3	Adaptive Threshold Setting	72
4.4	Performance and Discussion	73
4.4.1	Perceptual Robustness	73
4.4.2	Tamper Detection and Location	74
4.4.3	Source Camera Identification	75
4.4.4	PUF Reliability Discussion	77
4.4.5	Security Analysis	78
4.4.6	Comparison	80
4.5	Summary	81
5	Event-driven Dynamic Vision Sensor based Physical Unclonable Function for Camera Authentication in Reactive Monitoring System	83
5.1	CMOS Vision Sensor Preliminaries	85
5.1.1	Frame-based Active Pixel Sensor	85
5.1.2	Event-based Dynamic Vision Sensor	86
5.2	Proposed Event-driven PUF	88
5.2.1	New DVS Pixel Circuit	88
5.2.2	Process Variation Analysis	90
5.2.3	Sensor Architecture	91
5.2.4	Secure Transmission	94
5.2.5	Authentication Protocol	96
5.3	Performance Evaluation	99
5.3.1	Dataset Preparation	99
5.3.2	Ed-PUF Performance	100
5.3.3	Collision Test	103
5.3.4	Security and Overhead Analyses	106
5.3.5	Advantages over APS-based PUF	109
5.4	Summary	109
6	Conclusion and Future Work	111
6.1	Conclusion	111
6.2	Future Work	114
6.2.1	Event-driven PUF System Improvements	114
6.2.2	Aging Effect on PUF-based Unification Methods	115
	Publications	117

Table of contents

Bibliography	119
---------------------	------------

List of figures

2.1	Structure and response bit generation mechanism of memory-based PUF. . .	16
2.2	A classical code-offset structure based FE [5].	17
2.3	Reverse fuzzy extractor (a) Enrolment phase; (b) Straightforward authentication structure that has leakage; (c) Syndrome construction based authentication scheme. H^T is the transpose of the parity check matrix of a linear block code.	18
2.4	Structure of arbiter-based PUF.	19
2.5	A typical authentication scheme for strong PUF.	20
2.6	A typical sensor PUF with an ancillary conventional PUF.	23
2.7	Block diagram of CMOS image sensor based PUF.	24
2.8	(a) MEM relay structure; (b) A mass-spring-damper system balanced against an electrostatic force; (c) The delay-based pressure sensor PUF.	25
2.9	Operating principle of the user-device touchscreen PUF	26
3.1	(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.	37
3.2	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 $\tilde{T}_j^k$;	41
3.3	Distribution of v and its corresponding mean μ for each $m(m')$	45
3.4	Normalized Hamming distance distribution for (a) (genuine user, genuine device) (b) (fake user, genuine device) (c) (genuine user, fake device) combinations with varying m	46
3.5	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: $\mu = 0.497$, $\sigma = 0.015$	47
3.6	(a) Authentication accuracy with unreliable device keys; (b) FAR versus average bit error rate of genuine bio-codes.	49

List of figures

4.1	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.	65
4.2	An example of wrong matching: (a) original image I_o ; (b) received image I_r ; (c) recovered image I_t	67
4.3	Received image authentication of proposed data-device hashing.	67
4.4	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 is 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.	70
4.5	(a) e_{max} distribution for various cases of manipulations with different projection rates; (b) Enlarged details for the Y axis (e_{max}) range between $0 \sim 700$	71
4.6	Adaptive threshold determination.	72
4.7	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.	73
4.8	Examples of tamper detection and location for all manipulations shown in Table 4.2. (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.	76
4.9	Source camera identification performance.	77
5.1	Schematic of typical (a) 3T-APS and (b) 4T-APS for the frame-based image sensor.	86
5.2	(a) Pixel_o: Block diagram of a typical DVS pixel; (b) <i>Pixel_Core</i> : Transistor-level pixel schematic of a DVS; (c) Pixel_AER: Asynchronous event representation generator logic of the pixel.	87
5.3	(a) Operating principle of DVS. (b) 4-phase handshaking protocol to process the fired pixel [6].	87
5.4	New pixel design for the proposed Ed-PUF: (a) Transistor-level schematic; (b) Block diagram; (c) Simplified key component circuit for PUF mode of operation (the three added transistors are drawn in red).	89

5.5	Block diagram of the sensor architecture of the proposed DVS-based Ed-PUF system. (a) The PUF-embedded sensor chip with DVS event generation and PUF operation components. <i>Capacitance Change Detector</i> is used to detect severing or splicing of PUF response transmission line. The last modulo reduction block in <i>Challenge_Gen</i> is not required if the row and column number of the array is a power-of-two integer like 128, 256. (b) Key-gates and activation signals for logic obfuscation; (c) Load capacitance perturbation detector module; (d) Overview of the communications between the proposed sensor chip of (a) and an off-chip processor embedded within the DVS camera.	92
5.6	(a) The generation of an l -bit Ed-PUF response; (b) PUF-based HMAC for tag generation with syndrome construction based reverse fuzzy extractor incorporated to deal with the PUF reliability problem.	95
5.7	Authentication protocol of the proposed system. $\hat{R}$ is the noisy version of the correct R , which can be retrieved from the enrolled PUF database DB with the help of helper data h . tag_re denotes the recalculated tag. The command to “refresh C ” is denoted by dashed arrow since it may or may not be given depending on the rule set in the server.	97
5.8	Flow for activation and de-activation of logic obfuscation upon detection of load capacitance perturbation.	98
5.9	Uniqueness test ($\mu = 0.5005$; $\sigma = 0.0223$). The distribution of normalized HD is an approximate normal distribution.	101
5.10	Reliability tests against temperature and supply voltage variations.	101
5.11	Autocorrelation test result.	103
5.12	Construction of “event stream” frames by spatially composing the address-events collected (a) over a fixed period of time; (b) from a fixed number of events;	105
5.13	“Event stream” frames (composed by events collected over a fixed interval) made using (a) original event stream without PUF operation (video: https://youtu.be/n_JGVVi2nx8); (b) PUF-in-operation event stream (video: https://youtu.be/gDvOiX4f2c).	106
5.14	Layout of a single pixel using UMC 180nm technology with $C_1 = 240fF$ and $C_2 = 12fF$	108

List of tables

2.1	The Diehard and NIST statistical tests.	13
3.1	Mean μ_δ and standard deviation σ_δ of $\delta_{i,j}$ with varying m	45
3.2	Authentication performance with varying m	46
3.3	UDhashing discriminability test.	47
3.4	Verification performance	48
3.5	Comparison of user-device bipartite authentication schemes.	49
4.1	Perceptual robustness test.	74
4.2	Tamper detection tests.	75
4.3	System performance under unreliable R	78
4.4	Comparison with existing works.	80
5.1	Results of NIST tests.	102

Summary

With the boom of the Internet of Things (IoT), security is increasingly becoming a requirement rather than an option. Further development of IoT applications is challenged into seeking trust assurance of multiple credentials from user, device and data. Existing solutions meet this expectation with a naive layer-by-layer multi-factor authentication scheme. These solutions are typically realized using conventional techniques that require the safekeeping of a secret binary key in the non-volatile memory, or battery-backed SRAM of some device, or a token owned by the user, which have been proven to be vulnerable to various kinds of invasive, semi-invasive and side channel attacks. The non-repudiation assumption of these conventional techniques is being challenged by emerging artificial-intelligent assisted and malware-based attacks on independently chained weak links (factors), and their limited extension to IoT scenarios, where the sheer number of inexpensive and often insufficiently protected endpoints prevent application of standard, secret-key based techniques. A Physical Unclonable Function (PUF), which is a hardware-intrinsic security primitive emerged in the early 2000s, stood out as an inexpensive and yet effective way for device authentication and secret key generation. Compared to those conventional cryptographic primitives, the secret is hidden intrinsically within the device structure by the uncontrollable manufacturing process variations of integrated circuits. PUF promises a keyless solution with its appealing low-cost, low-power, tamper-aware and “reply-upon-request” attributes.

As the demand of providing more comprehensive protection across user, device and data for IoT applications is increasing, this research aims at *developing new PUF solutions to unify device, data and user authentication with high security and efficiency*. As a result of a comprehensive review of existing PUFs and multi-factor authentication schemes, new PUF based user-device hash, data-device hash as well as event-driven PUF design for end-point authentication, digital forensics and reactive monitoring are developed in this research.

The first distinctive contribution is a PUF-based user-device hash for endpoint authentication. A UDhashing scheme is proposed to unify the macroscopic human biometric and microscopic silicon entropy into a single identity by projecting the contactless facial biometric into a PUF-defined random space. The system achieves a bipartite authentication of both end user and end device as a whole, and zero-knowledge mutual authentication between the endpoint and the

verifier. The proposed system also achieves cancelable biometrics and system reconfiguration. In addition, the experimental results obtained using standard face databases and measured silicon data of a strong PUF demonstrates good authentication performance with excellent discriminability. The proposed system is also analyzed to be resilient against multiple attacks including memory probing, random guessing, replay, man-in-the-middle and modelling attacks.

The second contribution is a rotation/scaling-invariant PUF-based data-device hash for digital (more specifically, image) forensic applications. Existing digital forensics methods are capable of performing well in either forgery content detection or acquisition device identification but not both. At a time of rampant public media manipulation and ease of fake image/video fabrication, verification of benign and malicious image transformations is necessary but not sufficient for fact checking of visual artefacts presented in the court of law. The proposed scheme fills this gap by being able to detect and locate image tampering while identifying the source device that produces the images or footages with a comparatively low cost. The core idea is to create a hash tag to integrate the invariant image features with the PUF-based device “fingerprint” using random projection. The shaky security assumption of keeping a stored digital string (key) private for attestation in conventional perceptual image hash is avoided. For this use case, the proposed scheme can do away with challenge-response enrolment and hence the need for a trusted third party server to secure the database of challenge-response pairs. The system is demonstrated to achieve successful tamper detection and location, excellent data and device authentication as well as robust resilience against attacks on PUF.

Finally, the design of the first-ever event-driven PUF designed from the dynamic vision sensor (DVS) is proposed. DVS is an image sensor that responds asynchronously to relative changes in intensity. Compared with the conventional active pixel sensor, DVS reduces the data redundancy and possesses high dynamic range while preserving precise timing information. The event-driven PUF is derived from the original sensing circuit of DVS, with only three transistors added per pixel (originally 1 photodiode, 2 capacitors and more than 14 transistors) to achieve independent but simultaneous DVS sensing and PUF operation. The responses of the PUF is made to trigger only upon the detection of environmental changes. This is believed to be the only known image sensor based PUF whose response is indivisibly related to the sensing activities. This new subfield of event-driven PUF addresses indisputable attestation against tampering of hardware sensors and their generated data. It augments applications such as traffic monitoring and remote healthcare, by endowing the camera with a new security feature to generate a non-repudiable proof of exceptional events that have happened in surveillance. Experimental results show that the DVS sensor meets the quality criteria in PUF operation without compromising its normal imaging performance.

Chapter 1

Introduction

1.1 Motivation

The term “Internet of Things (IoT)” was coined by Kevin Ashton in 1999 [7]. The concept did not take off immediately until a decade later. The last decade especially has witnessed an explosive growth of connected smart devices. As predicted by Statista, the total number of connected devices will reach nearly 27 billion by the end of 2019, and 75 billion by 2025 [8]. This massive growth is driven by the significant yield improvement of CMOS integrated circuit (IC) manufacturing of miniaturized wireless devices, the increasingly powerful cloud computing network and the prevalence of integrated sensor technologies. IoT brings together people, devices, data and processes to make networked connections more relevant and valuable. It is undeniable that a more intelligent and connected world can bring many unthinkable benefits in efficiency, productivity, new infotainment experiences, faster and more value-added services, increased mobility, flexibility and accessibility, etc.. The associated threats, in particular the abuse that may lead to a security outbreak of this unprecedented scale of heterogeneity and density of connected devices, are equally unimaginable but disastrous. The connectivity and accessibility of IoT gives a local attack an exponential boost. For example, the leakage of a device identity may leak the physical locations and movements of multiple connected devices, which can be exploited for grievous intelligent terrorism and criminal activities. How far IoT can go and in which direction, boon or bane, depend largely on how secure it can be [9].

Fortunately, there is a global awareness of this emerging threat landscape. A number of proactive actions have already been taken to assuage IoT security. In October 2018, the department for digital, culture, media and sport (DCMS) of the United Kingdom government published the “Code of Practice for Consumer IoT security”, which released a set of guidelines with the aim of ensuring “secure by design” of IoT products [10]. European Telecommunications Standards Institute (ETSI) has also just released the first globally applicable standard

Introduction

ETSI_TS_103_645 for consumer IoT security in 2019 [11]. What is more, California's IoT law that requires all IoT device manufacturers to include reasonable security measures became effective on January 1, 2020 [12]. However, there is currently no consensus on what is the "reasonable security" that should be implemented in each different type of IoT device. Direct adoption of conventional cyber-security solutions is misleading in view of the completely different attack surfaces and physical constraints of IoT devices. Software-based methods for privacy and security protection are generally easier and more flexible to implement, maintain and update. On the other hand, hardware-based approaches can provide more direct and sophisticated system-level protection with dedicated trustable hardware supports. Though more costly and less flexible, the latter are generally more robust and less vulnerable than the former approaches. Their main benefits are listed below:

1. *Operating system (OS) independent security.* Software-based approaches are built on the assumption that the underlying lower layer application such as OS is well protected, which cannot be assured in most cases. Every security flaw in the OS, for example, memory content leakage through swap files, can easily compromise the security provided by the encryption codes. Hardware implementations utilize dedicated hardware security modules and therefore are better isolated and more immune from the inherent vulnerabilities of the OS.
2. *More costly to reverse engineering and power analysis attack.* Software-based approaches are merely instructions stored in accessible memory, which makes them much easier to be read, analyzed and modified to whatever requirements made by the attackers. Besides, the software commands have to be compiled into assembly language instructions with well-defined timing and power traces, which make them vulnerable to delay and power analysis side channel attacks. However, such attacks are more difficult to apply with high accuracy in hardware-based approaches without the assistance of more advanced measuring instruments and statistical modeling. Unlike software-based obfuscation, which is limited by their serial execution of instruction and fixed width arithmetic operations, hardware-based timing and power analysis obfuscation techniques proposed in recent years are more effective against general side-channel attacks.
3. *Improved speed performance.* Software implementations have to share the processing resources of the device, which lead to inevitable speed degradation of the system. To this end, digital signal processing (DSP) circuits are often utilized for faster multiplications. However, as DSP adopts publicly accessible buses for inputs and outputs, attackers are able to retrieve or even modify the temporary cryptographic values that are left on the bus between operations. Such vulnerabilities can be avoided by hardware-based approaches, whose cryptographic operations are off-loaded from the host system.

Though more superior than software-based implementation, hardware-based cryptography still suffers from the same dreadful drawback that the protection relies on the hard-to-solve mathematical problems with the assumption of a “safe” key. The local non-volatile storage of the secret key is the Achilles’ heel of many existing smart devices protected by established, classical cryptographic primitives. Cryptographic keys are relatively long random digital bit strings, which makes them to be well discriminated from the less entropic forms of memory data. They may be left intact in the memory even after their erasure [13]. As acutely commented by Ron Rivest, “calling a bit string a secret key does not make it secret, but rather identifies it as an interesting target for the adversary” [14]. The ease of attacking an IoT device was demonstrated by the Universal Forensic Extraction Device (UFED) developed by an Israel company named “Cellebrite”. With UFED, the physical/logical/file system extraction of all data and passwords (either intact, hidden or deleted) from a wide range of IoT devices, e.g., phones, portable GPS devices, tablets, smart watches and more, can be achieved within several hours [15]. The vulnerabilities stem from the insecure storage and transmission of personal and device identifiable information and the incapability of the latter to support strong encryption and use of a custom authentication process. Avoidance of the permanent presence of secret keys in vulnerable hardware is therefore imperative for lightweight and possibly mobile edge computing in IoT. As it is difficult to restrict an attacker’s accessibility to IoT devices, the solutions should at the very least, be tamper sensitive or tamper evident, if full tamper resistance is impractical.

The Physical Unclonable Function (PUF) is an emerging class of hardware-intrinsic security primitive that best fits the bill. The term was coined in 2002 [16, 17], and soon proliferated in the cryptography and security circles. Instead of relying on a hard-to-solve mathematical problem, the security of PUF rests in the intrinsic complexity and irreproducibility of a random physical disorder system. PUF tactfully exploits the circuit imperfections introduced by manufacturing process variations to generate a unique device signature. Benefited innately by greater variability of more advanced semiconductor fabrication process, PUF designs based on both conventional and emerging nano-scale ICs have permeated rapidly and dominate IoT security as a preferred root-of-trust for hardware identification. Such technology advancement makes efficient and compact integration with other cryptographic primitives in modern computing systems feasible without having to fully rely on the persistent presence of a secret key. The main advantages of PUF compared to conventional cryptographic methods are:

1. *PUF is unclonable, unpredictable, unique and reliable.* Thanks to the uncontrollable nature of manufacturing process variations, the characteristics of PUF cannot be physically replicated even by the original manufacturer with the same photolithography masks. Such characteristics make PUF the most suitable primitive for generating device signatures, which are similar to

Introduction

the human fingerprint. In addition, a “good” PUF is able to reproduce this device signature reliably even with varying temperature and supply voltage. This is a significant milestone for modern IoT applications that demand lightweight and reliable device authentication.

2. *PUF enables the device secret to be “reply upon request” and makes the secret tamper evident.* The secret information of the device is kept in its “response”, which can only be generated by querying the PUF device when it is powered on. The query is called the “challenge”. There is no need to store the secret locally in a non-volatile memory (NVM) like existing cryptographic methods. Besides, as the challenge to response mapping of the PUF is obtained through harvesting circuit process variations, any active manipulation of this disorder harvesting circuit will cause dysfunction of its challenge-response mapping mechanism and destroy the secret. Such “on-the-fly” key generation and “self-destruction” tamper-resistant properties substantially increase the barrier of invasive/semi-invasive attacks [18] in recovering the device secrets.
3. *PUF is flexible in design and has low power dissipation.* Randomness of conventional security primitives like Advanced Encryption Standard (AES) are typically realized in hardware by algorithm to architectural translation. Their secrets are represented and processed in digital form by hardware with modulo arithmetic components or lookup tables. In contrast, PUF circuits do not originate from high-level algorithms, but low-level analog parameter mismatches of nanoscale devices. Their stochastic challenge response mappings are achieved using specialized circuits derived from, for example, ring oscillator [19] or multiplexer chains [20]. It can also be built directly at no cost from existing circuit structure like SRAM [21] or image sensor arrays [22, 23] by exploiting their sense amplifiers to digitize the racing voltages of selected cells on power-up or reset. Today, most PUFs have been reported to consume power at μW level [24–26]. Spurious switching is avoided as the PUF operation is only triggered upon request. The vast circuit design options for harnessing the physical disorder with low-power characteristic make PUF a promising secure identity, secret protector and nonce generator for IoT devices.

As a hardware root of trust, PUF has opened up new horizons for solving IoT security problems. To date, PUF has received considerable attention not only from academia but also industry. There are increasingly new launches and repositories of security solutions, patents and products from startups and established companies, such as Intrinsic ID, ICTK, maxim, Xilinx, Qualcomm and QuantumTrace. For example, Xilinx has incorporated PUFs into its Zynq UltraScale+ MPSoC devices [27], NXP Semiconductor was the first company to integrate Intrinsic ID’s PUF into their smart card and embedded secure element chips [28], Qualcomm issued patent on delay-based PUF for masking memory-based PUFs to resist invasive and clone

attacks [29], and Accenture, a Dublin-based major consultancy firm, has partnered with Thales Group, a French multinational aerospace firm, to explore PUF with block chain technology for tracking, tracing and authenticating aircraft parts and materials [30]. As the technology matures, it is envisaged that new PUF structures will further vitalize and add versatile security features to broaden their application spectrum. Until 2019, the rise of IoT has created a huge influx of sensors and accelerated sensor standardization towards the ultimate aim of building a fully connected and cohesive supply chain. Sensors, as integral parts of an IoT ecosystem for life-changing applications, have always been a weak link and target of cyber-physical attacks [31]. Sensor-level authentication is an effective means to assure real end point security against the escalating sensor attacks and the imminent risk of data analytic attacks. Directly adding a PUF module to the sensor will put resource pressure on the sensor node, which usually has only limited footprint and energy budget. More importantly, the sensor-PUF interface where vital intermediate data are exchanged becomes susceptible as the sensing and security modules are separated. A better approach is to realize the PUF directly from existing sensor circuitry.

With sensors as the data feeder, IoT has certainly transformed our daily computing experience from siloed and specialized to ubiquitous and networked. A direct consequence of pervasive networked computations is the new gloss on existing interactions and distinctively new interactions between human and devices. The implication of an increasingly “data driven” networked society is “*any stand-alone or single-minded pursuit for security is no longer sufficient nor relevant*”. Therefore, to provide more comprehensive protection for IoT applications, it is important to endow PUF (especially sensor-based PUF) with the capability to not only identify the device, but also assure the integrity of the data it generated or acquired and authenticate the users who have privileged access to the device and its data. Although PUF, biometrics and other existing data analytic based security measures have been extensively studied and successfully applied in their own right, their unification to better address the new threat model and new use cases is still lacking.

1.2 Research Objectives

The research work described in this thesis aims at finding a solution *to expand PUF applications by developing new PUF-based solutions to fuse device, data and user authentication with high efficiency and security*. More specifically, the following research questions will be answered to address the shortcomings and application gaps of current PUF technology reported in the literature:

1. *Can device fingerprint and human biometric be coherently integrated without compromising performance to provide secure mutual authentication between end point and the server?*

Introduction

It is common to connect personal end-point devices like laptops, mobile phones and game play stations to the IoT network to access networked services and extra functionalities that are not available on the personal devices. In this scenario, authentication of the end point (including both end user and end device) is required to assure that the network resources are not abused or compromised, and the privacy of personal information should be well protected. Existing approaches usually treat either the user information, e.g., biometric features or password, or the device information, e.g., token or series number, separately and perform authentication on both end-points independently. Some approaches achieve both user and device authentication simultaneously with a sequential layered scheme. The security of such straightforward multi-factor methods is at best an aggregation of each single layer of protection. In most cases, it is only as strong as the weakest layer of protection when each independent layer can be attacked individually. As biometrics and PUF are unique identities of user and device, respectively, their fusion into an inseparable signature is much harder to break. Since human biometrics are largely immutable, it is important that the biometric template used to generate the unified signature is cancelable and unique for each application and will not be compromised by the authentication process or protocol. What is more, the end point should also have the ability to authenticate the service provider before any sensitive data is transferred.

2. *Can the acquisition device authenticity and media content integrity be validated after the media data have gone public?* The increasing number of deployed sensors has led to a large volume of sensor data being generated and transferred in the IoT network at every moment. The data could be generated from a malicious device or be replaced/modified during the transmission process. Therefore, before those data are utilized or analyzed to make any important decision or as a proof, it is important that the source of data acquisition and data integrity are validated against repudiation. An effective method would be to use PUF response as a birthmark of the sensor data. Since the sensor itself is resource constrained and generally has limited computing power, the PUF responses are preferably derived directly from the existing sensor circuit. An ideal imprint of a PUF response into its acquired (sensed) data should be able to authenticate the source device without leaking any device secrets, as well as detect and locate possible tampering of content information created by the sensor data without impacting the quality of perceivable content, nor falsely rejecting authentic content due to benign processing of sensor data.
3. *Can a PUF response be generated by an ad hoc event captured by the sensor in the field instead of a pre-enrolled challenge?* Existing PUF designs in the literature are mostly passive, i.e., its response is generated when stimulated by a challenge, but the challenge is either

pre-stored/generated in the device or received from the authentication server. This has limited its potential application against realistically fabricated evidence in digital forensics. For the use case against repudiation of evidence, it is more important to have the sensor PUF responds directly to its sensing data that has a direct semantic interpretation of the monitored events instead of being triggered remotely by user input. In other words, the real time events captured by the sensor should form an integral part of the PUF response generation mechanism. This will enable PUF to function as a proactive digital forensic component to collect and preserve evidence for identification of the occurrence of an incident. Such event-driven proof of reality is particularly useful to counteract anti-forensic techniques that attempt to erase or alter information, create pseudo-scientific chaff for spoofing, and implicate innocent parties by planting fake evidence. The main challenge in designing such an event-driven PUF is the ability to preserve the integrity and real-time performance of its original sensing function without intruding into the privacy of a monitored human subject and within reasonable cost and power consumption budget.

1.3 Major Contributions

The pursuit of the solutions to the aforementioned research problems has led to the following original contributions in this thesis: 1) A PUF-based User-Device hash for endpoint authentication; 2) A PUF-based Data-Device hash for digital forensics; 3) An event-driven PUF for reactive monitoring systems.

1.3.1 A PUF-based User-Device Hash for Endpoint Authentication

A novel UDhashing scheme capable of providing a bipartite authentication of both end user and end device as a whole, and zero-knowledge mutual authentication between the endpoint and the verifier has been proposed and demonstrated. This contribution aims to enhance the security of easy and fast biometrics-based log-in methods, which have replaced passwords to improve work efficiency and user experience in IT consumerization. In this work, a non-contact facial feature is extracted as the user identity. A strong PUF (see more details in Chapter 2.2) 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. Since the strong PUF has an inexhaustible number of challenge-response pairs (CRPs), fresh UDhashing identities can be generated to support cancelable biometric templates of the same user on different devices and for different services. Once an identity is compromised, it can be revoked by issuing another CRP for the generation of

a new 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 techniques without compromising security.

1.3.2 A PUF-based Data-Device Hash for Digital Forensics

This work particularly aims at solving the increasing rampant forgery problem of digital images and video footages. Digital forensics is challenged into seeking advanced technologies for forgery content detection and acquisition device identification. However, 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 are incapable of locating the tampered content of its captured images. This work proposes a new perceptual data-device hash 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 PUF-defined Bernoulli random space. The tamper-resistant random PUF response is unique for each camera and can only be generated when triggered by a challenge, which is provided by the image acquisition timestamp. The CMOS image sensor based PUF that is directly derived from the sensing element is utilized to generate the device signature. The proposed hash is evaluated on the modified CASIA database and the CMOS image sensor based PUF is 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.

1.3.3 An Event-Driven PUF for Reactive Monitoring System

One main role of surveillance footage is to provide a tangible proof of incident occurrence. Unfortunately, the assumed integrity of recorded video data and the authenticity of its originator cannot be assured by existing networked surveillance systems due to their accessibility and

the scale of cyberphysical threats. This problem is overcome by the proposal of an active PUF designed out of a dynamic vision sensor (DVS). DVS is an emerging video surveillance technology that outperforms traditional frame-based image sensors for its low latency, high temporal resolution and wide dynamic range. As DVS responds only to temporal intensity change and records only sparse asynchronous address-events¹ with precise timing information, it directly reduces the data size of activities and privacy intrusion of subjects under surveillance. This work introduces the first event-driven PUF system to fill the forensic gap of simultaneously authenticating the event data integrity and source camera identity by using a DVS camera for reactive monitoring. New DVS sensor architecture is proposed with negligible modifications. The Ed-PUF response bit can only be triggered by and uniquely dependent on the asynchronous addressed event without being interfered by the simultaneous firing of other address events. Address event streams are securely transmitted with an event package tag created by a keyed hashbased message authentication code with the key being the Ed-PUF response. A secure protocol to authenticate the identity of DVS camera and the integrity of address events transmitted through cellular network is also proposed. A camera lock is embedded to protect against severing and splicing the interchip connectivity within the camera for raw PUF responses. The proposed system is evaluated using raw baseline PUF data obtained by post-layout Monte Carlo simulation in UMC 180nm technology and real event stream captured by a DVS camera. The proposed Ed-PUF has been demonstrated to have excellent uniqueness, randomness and reliability. Collision analysis, testing and visualization are also conducted to show that the quality of DVS imaging is not compromised. Besides keeping the hardware/power/timing overheads low, the proposed scheme is also analyzed to be resilient against multiple attack scenarios.

1.4 Organization

The remaining contents of this thesis are organized as follows:

Chapter 2 presents a brief introduction of PUF and related works. The preliminaries of PUF including its definition, quality metrics (uniqueness, randomness, reliability) and corresponding evaluation methods are introduced. The widely adopted classification of PUFs, i.e., weak and strong PUFs, and their corresponding applications with required techniques or common vulnerabilities, i.e., secret key generation along with fuzzy extractors and device authentication along with modelling attacks, are presented. The chapter ends with a review of the emerging sensor PUFs and existing multiple factor authentication schemes in the literature.

¹Address-events or address events in this thesis refer to the DVS event data that contains the information of both event polarity and event address.

Introduction

Chapter 3 introduces the proposed design of UDhashing scheme for the generation of the user-device hash for endpoint authentication. This chapter begins with the review of the existing cancelable biometrics based on random projection and their bottlenecks. The detailed description of the proposed cancelable UDhashing scheme is divided into four core blocks, i.e., facial feature representation, device key generation, bio-code generation and recognition, and the UDhashing end-point mutual authentication scheme. The optimal selection of bio-code dimension, the discriminability/authentication performance of UDhashing and the device key reliability issues are also analyzed and discussed. Finally, a comprehensive analysis on the security level of the proposed scheme is also presented.

Chapter 4 presents a PUF-based data-device hash for tampered image detection and source camera identification. The preliminary background of image feature extraction methods and feature matching techniques, i.e., the speeded-up robust features (SURF) and the M-estimator sample consensus algorithm are introduced. The proposed PUF-based data-device hash is then presented where the rotation-invariant SURF features, adjoint block-based DCT features and CMOS image sensor based PUF are unified by means of random projection. This is followed by a detailed description of the proposed image tampering detection and source identification method. The performance of the proposed system is then evaluated and analyzed.

Chapter 5 is devoted to the event-driven dynamic vision sensor based PUF for camera authentication in reactive monitoring system. The fundamentals and differences in pixel structure of the frame-based and event-based vision sensors are introduced before the new DVS pixel circuit is presented. The process variation of the entropy source for this PUF response generation is analyzed. The corresponding sensor architecture and the secure transmission scheme are presented. A new authentication protocol is developed to perform device authentication and data integrity checking with a uniquely designed sensor lock. Finally, the performance (including event-driven PUF performance, collision analysis and test, security and overhead analysis) of the proposed event-driven PUF is comprehensively evaluated and analyzed.

Chapter 6 summarizes the major contributions of this thesis and outlines a few prospective future works that are worthy of further investigation.

Chapter 2

Literature Review

2.1 PUF Preliminaries

2.1.1 Definition

The silicon-based PUF was first proposed by Gassend et al. [16] as a physical random function in 2002. Around the same time, the optical PUF was also introduced by Pappu et al. [17] under the notion of physical one-way function. The term PUF is now commonly used to describe a diversity of circuit topologies that are developed to extract the parametric mismatches from the fabrication process variations of silicon devices for device authentication and secret key generation [19]. A native PUF can be formally defined as a physical system that maps a stimulating challenge c selected from a finite space $C = \{0, 1\}^k$ to an output l -bit response r in a finite space $R = \{0, 1\}^l$ according to its intrinsic stochastic properties induced by its device manufacturing process as follows:

$$PUF : C \rightarrow R \quad (2.1)$$

The operation of generating a response $r \in R$ given a challenge $c \in C$ can also be mathematically expressed as:

$$r \leftarrow PUF_{\alpha_k}(c) \quad (2.2)$$

where α_k represents the disordered physical characteristics of the PUF instance PUF_{α_k} .

2.1.2 Evaluation Metrics

ISO/IEC DIS 20897 [32] has specified the security requirements and test methods for PUF for generating non-stored security parameters from an industry standpoint. For academic community, the following desiderata are widely considered in the design of PUFs.

Literature Review

- *Reliability*: A PUF is reliable if its response to the same challenge can be reproduced over time and over a wide range of operating conditions.
- *Unpredictability*: The response to an arbitrary challenge generated from a PUF cannot be predicted from the Challenge-Response Pairs (CRPs) of another PUF instance or from the used CRPs of the same PUF instance.
- *Unclonability*: The CRP mapping of a PUF is unique and cannot be physically or mathematically cloned. This holds true even for the original manufacturer of the PUF.
- *Physical Unbreakability*: A PUF is tamper evident such that any physical attempts to maliciously modify it will result in malfunction or permanent damage to the chip.

However, to accurately measure or quantify the above quality criteria is not always possible. Typically, a PUF is evaluated for the uniqueness, randomness and reliability of its CRPs by the following statistical methods.

Uniqueness

The response to a challenge generated by a PUF instance should be unique and distinguishable from those generated by other PUF instances when the same challenge is applied. The uniqueness of a PUF can be estimated statistically by the Hamming distance between the response bit strings generated from different PUF instances to the same challenge as follows,

$$Uniqueness = \frac{2}{I \times (I - 1) \times M} \sum_{k=1}^M \sum_{i=1}^{I-1} \sum_{j=i+1}^I \frac{HD(R_{i,k}, R_{j,k})}{l} \quad (2.3)$$

where I is the number of sampled instances, M is the number of sampled responses from each instance and l is the bit-length of a string. $HD(a, b)$ is a function that evaluates the Hamming distance between two binary strings a and b [33].

The uniqueness has an ideal expected value of 0.5, and the ideal standard deviation σ for a l -bit response can be calculated as [34]:

$$\sigma = \frac{\sqrt{l}}{2l} \quad (2.4)$$

Randomness

Ideally, a PUF should have an equal probability of generating response bit '1' and response bit '0', i.e., the response bits generated by a PUF should be uncorrelated so that the bit generated

Table 2.1 The Diehard and NIST statistical tests.

Test No.	Diehard	NIST
1	Birthday spacing	Frequency (Monobit) Test
2	Overlapping permutations	Frequency Test within a Block
3	Runs test	Runs Test
4	Monkey tests	Test of the Longest Run of ‘1’ in a Block
5	Count the 1s	Binary Matrix Rank Test
6	Parking lot test	Discrete Fourier Transform (Spectral)
7	Minimum distance test	Non-Overlapping Template Matching Test
8	Random spheres test	Overlapping Template Matching Test
9	The squeeze test	Maurer’s “Universal Statistical Test”
10	Overlapping sums test	Linear Complexity
11	Ranks of matrices	Serial Test
12	The craps test	Approximate Entropy Test
13	-	Cumulative Sums Test
14	-	Random Excursions Test
15	-	Random Excursions Variant Test

from a PUF cannot be predicted by knowing its challenge. The entropy of a discrete random variable X with probability $p_x = P_r(X = x)$ is defined as [22]:

$$H(X) = - \sum_{x \in X} p_x \log_2 p_x \quad (2.5)$$

However, as the actual distribution of CRPs is unknown, it is difficult to measure the entropy directly. Alternatively, minimum-entropy estimation of the generated PUF response bits has been explored for randomness evaluation [35–37]. In addition, autocorrelation test and test suites such as diehard [38] and NIST (National Institute of Standards and Technology) [39] are also commonly used to evaluate the randomness of a PUF¹. Autocorrelation is diagnosed using a correlogram and the test shows how data is correlated with itself [4]. The diehard and NIST tests both consist of a battery of statistical tests (see Table. 2.1 for details). For NIST test suite, each test will output its corresponding *P-Val* and a proportion rate *PROP*. For the tested response bit stream to be considered as random, the *P-Val* of each statistical test should be larger than 0.01 and the *PROP* should exceed certain threshold. For N samples of bit sequences, the *PROP* threshold T is calculated by [40]:

$$T = (1 - \alpha) - 3\sqrt{\frac{\alpha(1 - \alpha)}{N}} \quad (2.6)$$

¹It is usual to have more than one set of tests conducted to prove the randomness of PUF.

where α is the significance level 0.01. If $N=100$, then $T=0.96$, which means at least 96 out of 100 ($PROP \geq 96/100$) test sequences should pass this statistical test.

Reliability

The responses to the same challenges generated from a PUF at different time under different environmental conditions should be identical. The reliability of a PUF is evaluated by the bit error rate (BER). The BER of a PUF can be calculated by the intra-die Hamming distances, which is determined by the mean number of flipped bits among the regenerated responses at different time instants, i.e.,

$$Reliability = (1 - \frac{1}{m} \sum_{t=1}^m \frac{HD(R_i, R_{i,t})}{l}) \times 100\% \quad (2.7)$$

where m denotes the number of sampled responses. R_i is a l -bit response produced by a PUF instance i under the nominal operating condition and $R_{i,t}$ is the response reproduced at time t under possibly changed operating conditions. Ideally, the reliability should be 100% under the specified range of operating voltage and temperature.

Note that if the PUF response is used as a secret key for encryption or decryption operation, 100% reliability is required. This is difficult to achieve in practice. As a result, error correction codes (ECCs) are necessary to ensure the correct regeneration of PUF responses. Repetition codes [41], Hamming codes [42] and Bose-Chaudhuri-Hocquenghem (BCH) codes [43] are some of the widely used ECC techniques. The repetition code is the simplest ECC technique but has relatively high redundancy in its codeword. There are some PUF implementations that utilize repetition codes for error correction [44]. Hamming code, on the other hand, has the highest code rate with the minimum distance of three. Hence, it can only correct one bit error or detect up to two bit-bit errors. Due to the very low error correction rate, Hamming code is the least popular choice for PUF response reconciliation [45]. The BCH code is most widely used by PUF developers. BCH codes are capable of correcting multiple bit errors and have high flexibility in code parameter selection. Therefore, custom codes can be designed given different specifications. What is more, the decoder design of BCH codes is very simple, which enables them to be easily implemented in small and low-power electronic devices [46]. For example, the BCH (127, 64, 10) code is able to correct up to 10 errors in a 64-bit message (PUF response) with the code block length being 127 bits.

2.2 PUF Categories

Many new and different implementations of PUF have been proposed to harness the entropy source of their device manufacturing process variations. It becomes increasingly difficult to categorize different types of PUF based on their mismatched analog parameters (e.g., delay, power, voltage, current, etc.) or their circuit topologies (e.g., arbiter, memory, ring oscillator (RO), etc.). As the size of CRP space has a direct implication on its applications and their threat model, the existing PUFs are often dichotomized into weak PUF and strong PUF [47, 48].

2.2.1 Weak PUF

Weak PUFs have limited CRPs. More specifically, the number of CRPs increases linearly or polynomially with the number of basic cells or symmetric component blocks abutted to form a PUF. In the worst case, there is only one CRP for a PUF [17]. The CRPs are dependent merely on the physical disordered properties of individual devices of each basic cell of the PUF, and possess no (or negligible) correlation with each other. More precisely, the CRPs of a weak PUF with finite physical size can be exhaustively measured within polynomial time. Because of that, the peripheral interface of a weak PUF must be protected by integrating it with a fuzzy extractor (FE) to restrict direct access to the original response generated internally by the native PUF.

The external output response r due to the FE $Ext()$ can be expressed as:

$$r \leftarrow Ext_{\alpha_e}(PUF_{\alpha_k}(c) + e, h) \quad (2.8)$$

where e is the error bit string induced by noise, $h \in H$ is the helper data used by an ECC to correct the noisy responses of the native PUF, followed by a universal hash function to enhance the output response entropy. $\alpha_e \in E$ is an attribute of ECC and the hash function used in $Ext()$. With the FE, the CRP mapping becomes

$$PUF_E : C \rightarrow R \times H \quad (2.9)$$

Examples of weak PUFs include most memory-based PUFs and the coating PUF [49]. Normally, a memory-based PUF exploits the metastable states of cross-coupled inverters to generate random bits during reset [21, 50–55]. As shown in Fig. 2.1, the response bit is determined by which of the two identically sized inverters in a memory cell addressed by the challenge bit reaches its trip-point faster. Variants of memory-based PUFs have been developed in the last few years, such as latch PUF [52], butterfly PUF [50], SRAM PUF [21], D flip-flop PUF [56] and bus keeper PUF [51]. The cell structure of each type may be unique

Fig. 2.1 Structure and response bit generation mechanism of memory-based PUF.

but the generated random response bit is still independently read out from each cell of the array addressed by the challenge. The memory-based PUF is easy to design, has low area overhead and is devoid of mutually correlated response bits. Being a weak PUF, its challenge-response space can be entirely explored with trivial effort [48], which compromises its security as a low-cost authenticator.

From the application perspective, weak PUFs are normally used in cryptographic key storage/generation [37, 54, 55, 57], device identification [58] and brand name and IP protection [50, 59]. The application of weak PUF alone is inadequate for authentication unless it is used along with a keyed hash message authentication code (HMAC) [60]. A helper-less PUF was proposed in [61] to allow two parties to authenticate each other by sharing a short-term instead of a long-term secret in an Encrypted Key Exchange (EKE) protocol [62], with an encryption function and a cryptographic hash [63] to generate the transitory secret.

Fuzzy Extractor and Reverse Fuzzy Extractor

A FE is typically used for the generation of cryptographic key K and helper data from the raw PUF response, and for the recovery of K from the noisy PUF responses with the helper data. These two phases of operation are shown in Fig. 2.2. In the generation phase ($Gen()$), a uniform random codeword C_s is picked to calculate the offset h from the noisy PUF response R . A hash function h_σ is also randomly selected from a universal family, and applied to R to generate the secret key K . Both the offset h and the hash function index σ are helper data, which can be stored in public as it can be constructed in a way that will not compromise the privacy of the secret key (in an information-theoretic sense). In the reproduction phase ($Rep()$), when the same challenge is applied to the PUF, its corresponding helper data will be retrieved

Fig. 2.2 A classical code-offset structure based FE [5].

and supplied to the FE to regenerate the secret key. As the key can always be generated on demand, there is no need to store the key in memory, which minimizes the risks of attack that long persecuted EEPROM-based key storage in computers.

However, though important, FE is not very suitable for many PUF-based applications since the underlying error decoding module in the reproduction phase is typically resource-intensive, especially when multiple bit errors in the PUF responses have to be corrected. This problem can be solved by the reverse fuzzy extractor (RFE) [64], whose key idea is to offload the computational-intensive $Rep()$ to the more powerful server side. It is important to note that the secret that the prover wants the verifier to validate with the RFE is the noisy PUF response R' instead of the enrolled PUF response R in the conventional fuzzy extractor. Fig. 2.3(a) shows the enrolment phase of the reverse fuzzy extractor, which requires the server (verifier) to only collect the PUF response R pairs. No helper data is required to be generated at this time. In the authentication phase shown in Fig. 2.3(b), after a potentially noisy PUF response R' has been generated, a helper data h will be generated by $Gen(R')$. This helper data h is then sent to the verifier. The verifier can recover R' by executing $Rep(R, h)$. However, RFE has the concern of leaking secret information. Each execution of $Gen(R')$ on a different noisy version of the same PUF response reveals a new helper data, the attack is able to figure out the secrecy if he/she knows the error pattern e between R and R' [65, 66]. In PUF-based applications, the error pattern e is determined by the inherent noisiness of a PUF. For this reason, the helper data may reveal new information about the PUF response if its BER is biased. For RFE to be used securely, the PUF should have a balanced BERs of bit '0' and bit '1'. Therefore, though the syndrome construction based RFE shown in Fig. 2.3(c) is claimed to be reusable in [64], the

statement may not hold when BER of the underlying PUF response is dependent on its enrolled bit value [66]. As observed in [66], for a PUF with the probability of generating bit ‘1’ falls in the range of [0.42, 0.58], the extra min-entropy loss resulted from the leakage of bit-specific reliability information is very small. If unbalanced BERs are observed, either redesigning the PUF or using de-biasing techniques can be utilized to mitigate the problem. Besides, the work [67] also proposed methods to deal with the secrecy leakage problem of RFE.

Fig. 2.3 Reverse fuzzy extractor (a) Enrolment phase; (b) Straightforward authentication structure that has leakage; (c) Syndrome construction based authentication scheme. H^T is the transpose of the parity check matrix of a linear block code.

2.2.2 Strong PUF

The strong PUF, on the other hand, is characterized by a huge number of CRPs, which grows exponentially with the number of symmetric component blocks used to create the PUF. This property makes it practically infeasible to exhaustively evaluate all the challenges within a realistic time. Owing to its physical obfuscated structure and complex combinations of component mismatches, the probability of correctly predicting the response to a randomly chosen unknown challenge is very low even with the knowledge of many other CRPs. For these reasons, a strong PUF does not have to encapsulate a post-processing module to avoid its originally generated response bits from being accessed externally. This attribute makes strong PUFs well suited to a wide spectrum of security applications and advanced authentication protocols without routine computational assumptions or the need for additional cryptographic hardware. The irresistible progress of CRP generation does not, however, elude the responses of strong PUF from being potentially corrupted by noise. The random and unpredictable noise

Fig. 2.4 Structure of arbiter-based PUF.

inherent in the response turns out to be useful in some advanced protocol construction if the strong PUF is augmented by some helper data to provide controlled redundancy.

Examples of strong PUF include optical PUF and various types of arbiter-based PUF such as simple arbiter PUF [20], lightweight secure (LW) PUF [68] and XOR arbiter PUF [69]. Delay based strong PUFs typically generate their CRPs by establishing a race condition in symmetric chains of identically designed components to generate random response bits. A simple arbiter-based PUF consists of a daisy chain of cross-bar switches as shown in Fig. 2.4. An input transition is launched simultaneously into an upper path and a lower path. The signal in each path is fed to the input of one of the two multiplexers in each stage controlled by a challenge bit. Although the two selected paths are laid out in an identical fashion, the gate and net delay variations will result in a subtle timing difference between them. The D-latch acts as an arbiter to decide the output response based on the winner of the race. While the large CRP space of arbiter PUF is immune to brute-force attacks, the correlated CRPs are vulnerable to machine learning (ML) attack. By training a properly selected machine learner with CRPs collected from an arbiter PUF, unknown responses generated from the PUF can be correctly predicted with high confidence [48] (See more details in “**Machine Learning Based Modelling Attacks**”).

Strong PUFs can be applied in more authentication protocols at lower cost. Fig. 2.5 shows a typical PUF based authentication scheme between a trusted central authority A and a party B . A collects a set L of CRPs (c, r) generated from the PUF token P and stores L in a database for further interrogation. P is then delivered to B so that B can use P for subsequent credible communications with A in a public network. To authenticate P , A randomly selects an unused challenge c_i and sends it to B . B applies c_i to P to generate a response r'_i and sends it back to A . A validates r'_i against L . If the CRP matches, then B is in possession of an authentic P .

The above protocol assumes that only the trusted party A and the person B in possession of the authentic token P can produce the correct CRPs. Because of the exponentially large

Fig. 2.5 A typical authentication scheme for strong PUF.

CRP space, the binary channel used for the interrogation of challenges and responses can be non-confidential. An untrusted but reliable physical delivery channel for the transportation of P is hypothesized in which an adversary can have read-only access during the manufacturing process or transit of P through the channel. To prevent the PUF from falling into the hands of an adversary who can use it directly to impersonate B , acknowledgment of its receipt by the intended recipient B through a private channel should be received within a reasonable stipulated time before P can be used for subsequent authentications through the unprotected public channel. To avoid replay attacks, a unique challenge should be used for each round of authentication. Therefore, it is important that A stores a sufficiently large number of CRPs and periodically refreshes the CRP database [70].

Machine Learning Based Modelling Attacks

Machine learning (ML) based modelling attacks have been a major threat to strong PUFs. Many recent experiments and research works have shown that certain strong PUF structures can be mathematically modelled and cracked by ML algorithms. This kind of attacks targets strong PUFs with a large number of CRPs [60]. Modelling attacks are usually implemented by deriving a numerical simulation model of the PUF. A ML algorithm is then used to predict its output responses to unknown challenges by training the machine learner with a subset of known CRPs. Those CRPs can either be collected from direct access to PUF or by eavesdropping the protocol.

Take the standard arbiter PUF as an example. The four possible propagation delay paths through each crossbar switch with two-input ports and two-output ports can be represented by δ_{tt} , δ_{bt} , δ_{tb} and δ_{bb} as shown in Fig. 2.4, where the subscript t and b denote the top and bottom

ports of the switch, respectively. Let $C[i] \in \{-1, 1\}$ denote the bipolar-encoded challenge bit of the i -th switch. Assume that the i -th switch operates in straight mode when $C[i] = 1$ and swapped mode when $C[i] = -1$, then the path delays, $\Delta_t[i]$ and $\Delta_b[i]$, of the launching pulse from the input ports of the first switch to the top and bottom output ports, respectively, of the i -th switch can be modelled by:

$$\begin{cases} \Delta_t[i] &= \frac{1+c[i]}{2}(\Delta_t[i-1] + \delta_{tt}[i]) \\ &+ \frac{1-c[i]}{2}(\Delta_b[i-1] + \delta_{bt}[i]) \\ \Delta_b[i] &= \frac{1+c[i]}{2}(\Delta_b[i-1] + \delta_{bb}[i]) \\ &+ \frac{1-c[i]}{2}(\Delta_t[i-1] + \delta_{tb}[i]) \end{cases} \quad (2.10)$$

Let $\Delta[i] = \Delta_t[i] - \Delta_b[i]$. For an n -stage PUF, the delay difference between the top and bottom paths at the input of the arbiter can be derived as [71]:

$$\Delta[n-1] = \vec{\omega} \cdot \vec{\phi}^T \quad (2.11)$$

The $n+1$ linear additive model coefficients $\vec{\omega}$ of the delay vector are calculated by:

$$\omega[i] = \alpha_i + \beta_{i-1}, \text{ for } i = 0, 1, \dots, n. \quad (2.12)$$

where $\alpha_n = \beta_{-1} = 0$ and $\alpha_i = \frac{1}{2}\{(\delta_{tt}[i] - \delta_{bb}[i]) + (\delta_{tb}[i] - \delta_{bt}[i])\}$ and $\beta_i = \frac{1}{2}\{(\delta_{tt}[i] - \delta_{bb}[i]) - (\delta_{tb}[i] - \delta_{bt}[i])\}$ for all $i = 0, 1, \dots, n-1$.

The $n+1$ elements in the feature vector $\vec{\phi}$ are determined by:

$$\phi[i] = \begin{cases} 1 & i = n, \\ \prod_{j=i}^{n-1} C[j] & i = 0, 1, 2, \dots, n-1. \end{cases} \quad (2.13)$$

The response of an n -stage arbiter PUF can thus be deduced directly from the sign of $\Delta[n-1]$. It was reported in [72] that a few hundred CRPs are sufficient to achieve a predication rate very close to the reliability of the attacked PUF using this model.

As a countermeasure against ML attacks, nonlinear effect can be added, for example by XORing the responses of multiple arbiter PUFs. A study was carried out in [73] to attack standard, XOR, LW and feed-forward arbiter PUFs by different ML algorithms trained using simulated, noise free subset of CRPs as well as real silicon data from the first two types of PUFs. It was found that while strong PUFs are generally insecure against advanced ML attacks, the complexity of the attacks can be easily increased by increasing the number of

stages, the number of arbiter chains or by adding new nonlinear elements. For example, the ML resistance increases polynomially with the number of stages (or equivalently, bit length of challenge) but exponentially with the number of arbiter chains of XOR and LW arbiter PUFs. The effectiveness of the latter comes at a price of degraded response stability. A 512-stage XOR arbiter PUF with 8 arbiter chains was found to strike a good balance between the number of noisy responses and silicon premium to effectively thwart ML based modelling attacks [73].

Another widely conceived approach to defend ML attacks is to hide the direct responses and transmit either the helper data or substring\padded substring of the response through the authentication protocols, as exemplified by the RFE protocol [64] and Slender PUF (SLPUF) protocol [74, 75]. However, the assumption that ML attacks can only be successfully conducted with knowledge of native CRPs is found to be fallacious. It was demonstrated that evolution strategy based ML technique is able to break the RFE and SLPUF protocols without direct access to CRPs but using helper data (or unreliability information leaked by helper data) of ECCs and substring\padded substring of responses, respectively [76]. Using PUFs that are more ML resistant can help mitigate this problem.

2.3 Related Works

2.3.1 Sensor-based PUF

Sensors play a key role in data collection from environments. As the collected data are often used for analysis and decision making, a sensor is capable of crippling a large networked infrastructure if it is hijacked to inject bogus data. A common approach to secure a sensor node is to incorporate an external PUF instance into the sensing system, as exemplified by the authentication of smart meters in a smart grid system [77–79]. In these applications, the sensor output can either be used for generating PUF challenges [80] or be encrypted by PUF responses [81, 82]. A generic sensor PUF architecture to interwind conventional silicon PUF responses into sensor data is proposed in [81], which is demonstrated by an array of photodiodes. The photodiodes are coated with non-uniform optical transmittance to affect their optical sensitivity randomly. The photodiodes are grouped in three and the slope of the approximately linear response function generated by one of the three photodiodes is compared against its offset derived from the other two photodiodes to produce a DC voltage. This DC voltage is dependent on the optical sensitivity of the photodiodes. As shown in Fig. 2.6, the currents from a subset of photodiode groups are selected into two summing circuits. The summed currents, P and Q , are compared to generate a raw binary response. The summation is controlled by a stream cipher, which is initialized by the response of a conventional PUF with an input challenge.

Fig. 2.6 A typical sensor PUF with an ancillary conventional PUF.

The sensor itself can also be turned into a PUF without compromising its sensing capability. For example, in the absence of illumination, or in uniform illumination, device mismatches and offset due to the manufacturing process variations can cause the individual photodiode of a CMOS imager to respond differently from pixel to pixel, giving rise to random fixed pattern noise (FPN). As FPN can badly degrade the image quality, correlation double sampling (CDS) is used in commercially available CMOS cameras to cancel the reset voltage from the signal voltage after exposure. Because of the CDS, the FPN cannot be directly accessed from the pixel readout circuit. To overcome this problem, Cao et al. [22] substituted the CDS by a column-level CDS activation and bypass circuit to switch between regular sensing and PUF modes of operation. During normal exposure, the sensor amplifier outputs the difference between the reset and signal voltages. During the PUF mode, the reset voltage is directly read out by turning on the CDS bypass switch to prevent the signal voltage from biasing the FPN and impacting the randomness of the PUF response. The digitized sensor outputs are buffered and fed into the CRP generator to produce the response bits, as shown in Fig. 2.7. As the reset voltage is sensitive to random reset noise and variations of supply voltage and temperature, the device signature generated may not be reproducible. To obtain a more reliable signature from the image sensor, a differential readout scheme is used to extract the response bit from the reset voltages of two pixels selected by the outputs of a linear feedback shift register (LFSR) controlled by an input challenge. When the difference between the two reset voltages is larger than an empirically determined threshold P_{th} , the generated bit will be retained as the response bit to the input challenge. Otherwise, another pair of pixels will be sought by shifting the

Fig. 2.7 Block diagram of CMOS image sensor based PUF.

content of the LFSR until a stable CRP is found or all pixels of the entire reset image have been exhausted. One advantage of this proposed solution is that the entire CRP mappings of the PUF can be reconfigured by selecting a different seed value to initialize the LFSR. This is particularly useful when the original CRP mapping is suspected of being compromised or the CRP can be periodically refreshed to thwart modelling attacks.

Pressure sensors can also be authenticated by generating PUF responses from the pressure sensitivity of an electrostatically actuated micro-electro-mechanical (MEM) relay [83]. Figs. 2.8(a) and (b) show the basic structure and lumped model of a four terminal folded flexure MEM relay. It consists of a moveable gate suspended by four spring-like flextures above the body, drain and source electrodes. When a voltage is applied between the body and gate electrodes, the electrostatic force will pull the gate towards the body electrode against the spring force. The relay turns on when the applied voltage is large enough to move the gate through the air gap channel to form a conductive path between the source and drain electrodes. The gate will be released to its original floating status when the applied voltage is sufficiently reduced [84]. The mechanical delay t_{mech} was proven to dominate the relay delay characteristic

Fig. 2.8 (a) MEM relay structure; (b) A mass-spring-damper system balanced against an electrostatic force; (c) The delay-based pressure sensor PUF.

[85] and has positive and negative correlations with the ambient air pressure and gate overdrive voltage, respectively. For a fixed gate voltage, the body electrode can be biased to make an “N-type” or a “P-type” relay [86]. These phenomena enable a variable speed inverter to be built with a MEM relay. Fig. 2.8(c) shows a RO PUF built by MEM relay inverters with a bias generator for changing the inverter speed [83]. The response bit is generated in two phases. In the first phase, one oscillator is triggered by the counter to measure the pressure in a regular interval. Based on the measured pressure, the bias generator produces a set of control voltages to keep all but one relay inverter in each oscillator in “high-speed” mode so that the oscillation frequency is dominated only by the “low-speed” inverters. In the second phase, a response bit is generated by comparing the frequencies of two ROs selected by the challenges. This MEM-relay sensor PUF can also be used as a sanity check to conditionally generate the obfuscated key for validation when the sensor node is operating in unusual pressure ranges.

2.3.2 User-Device Authentication

As the consumerization of IT grows, an increasing number of network environments are faced with the influx of external or remote users using personal devices. The necessity of authenticating both end-point user and end-point device has been gradually proposed by commercial companies. Though named differently, configuration managers like “End-User Device Registration” by VMware [87], “User Device affinity” by Microsoft [88], and “User Device Tracker” by SolarWinds [89] have been established with a common aim to support commercial remote centers to automatically link each authorized end-user with one or several

legitimate end-devices in order to detect and deny access when rogue devices and users are trying to connect to the network.

In the literature, authentication of both user and device is mainly explored in patents [90–92]. Such methods usually adopt a separate two security layer structure, i.e., perform user or device authentication in a sequential manner. A substantial number of messages have to be transmitted and operation delay is introduced inevitably. To protect the sensitive device/user credentials during transmission, encryption keys are usually required and have to be stored in the end device. Therefore, those embodiments are still vulnerable to the pre-discussed NVM key retrieval attacks using invasive or semi-invasive techniques.

Fig. 2.9 Operating principle of the user-device touchscreen PUF

A totally different user and device authentication scheme has emerged, as exemplified by [93]. In this work, a new notion of user-device (UD) PUF is proposed. Later in 2017, the same research group proposed another implementation of this concept in [94]. The key idea behind [93, 94] 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. Fig. 2.9 shows how the concept is realized in [93] by the touchscreen of a mobile device. The challenge of the UD-PUF is a series of binary coordinates $(X_i, Y_i) \cdots (X_j, Y_j)$ that forms a shape on the touchscreen for the user to trace, and the response is a digital word obtained by quantizing the sequence of sensed pressure values read from an Android APP. Arguably, the finger pressures asserted on touchscreen in tracing two adjacent points are correlated but not the sensitivities of two adjacent sensing elements. In such methods, the desirable intrinsic parametric changes contributed by device fabrication process variations are not structurally harnessed, which makes them highly susceptible to 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 the user biometric whereas noise reduction in biometric information processing will distort device parametric distribution.

Another “UD-PUF” was the very recent work [95], which 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 noise aware - interval optimized mapping bit allocation (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 will be compromised.

2.3.3 Data-Device Authentication for Digital Forensics

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. Their easy-to-access and low-cost attributes, however, also attract attention from attackers. The concerns of image security are escalating. There are two main problems to be solved, i.e., detection of image tampering and authentication of the imaging device of the image in question. These two problems are usually tackled independently, and intensive research efforts have been expended in each individual problem in the past decade.

For image tampering detection, the solutions are mainly provided by three types of schemes in the literature [96]: image watermarking [97, 98], digital image forensics [99–102] and perceptual image hashing [103–105]. 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 a 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 [106]. Digital image forensics based schemes aim at blind investigation of malicious tampering 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 [107]. 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. As discussed, storing the secret key in a NVM is vulnerable to data remanence and reverse engineering attacks [108–110]. 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 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 the 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 [111]. Most existing works focus on imaging device brand identification [112, 113], 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 [114–118], 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, which makes those schemes vulnerable. In contrast, CMOS image sensor based PUF [22] (as introduced in Sec. 2.3.1) presents a newer and anti-forensic resistant way of individual camera device identification. Instead of extracting device features from the images, CMOS image sensor based PUF directly extracts the “device signature” from the pixel array [22]. Their proposed low-cost CMOS image sensor based PUF was 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. This work has been demonstrated to be able to differentiate every single camera device regardless of model or brand with high accuracy.

2.4 Summary

In this chapter, some background information about PUF is introduced. A comprehensive review of sensor-based PUF and multi-factor authentication, i.e., user-device authentication

and data-device authentication in digital forensics, is also presented. Our study corroborates that PUF is a promising low-cost primitive for IoT device identification. The design flexibility of PUF enables it to be implemented directly by the original sensing system without incurring high hardware overheads. Our survey also reveals that existing authentication schemes of user, data and device are not inclusive. Silo approaches are not efficient, as they usually require expensive conventional encryption/decryption methods to protect the authentication credentials. The inevitable local storage of cryptographic keys are vulnerable to attacks. The idea of keyless PUF-assisted multiple factor authentication has recently been toyed with. Unfortunately, the fusion of multiple factors of authentication with PUF in this nascent stage has poor detection accuracy and impractical constraint in implementation. In the following chapters, several new PUF integrated security solutions that securely and efficiently unify user and device authentication (Chapter 3), data and device authentication (Chapter 4), data-triggered data and device authentication (Chapter 5) have been developed.

Chapter 3

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

As COVID-19 pandemic is escalating from epidemic to pandemic and spreading all over the world by March 2020 [119], multiple cities and even countries (Italy, Spain, Malaysia, France, and so on) are forced to lockdown. Employees and students are advised to work and study from home to prevent further spreading of the contagious virus [120]. Thanks to the technology advancement, personal devices like laptops, tablets and smart phones today are powerful enough to fulfil the current urgent demands. Work and study from home policies are popular and commonplace even in normal time due to the more advanced and updated personal devices than the IT equipment owned by the companies or schools. The emerging trend of Bring Your Own Device (BYOD) thrives as more organizations buy into this idea. BYOD systems allows the employees to connect their personally held devices to the inner corporate network without the restriction of time and place. This developing trend is irresistible with appealing good public hygiene, 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. Therefore, measures to ensure that only an authorized person with trusted devices can access the corporate networks are of paramount importance for enterprise to contrive BYOD.

To design a robust endpoint authentication system, three objectives should be fulfilled, i.e., 1) *easy and fast log-in methods* like a glance, a touch or an utter of speech are preferred in order to improve work efficiency and user experience; 2) *the system should have a high security level*. Storing any secret credentials such as a password or an encryption key is unsafe as digital content in non-volatile storage is known to be recoverable by invasive and semi-invasive

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

attacks [1] or even mature products [15]; 3) *user privacy needs to be protected* to refrain the endpoint from giving out sensitive information to a masquerade system. As reviewed in Sec. 2.3.2, existing authentication schemes are incapable of fulfilling all three objectives, the last objective in particular. Most of them [90–92] adopt separate security layers for user and device authentication and require encryption/decryption techniques with secret keys to be used to protect authentication credentials. Only a few PUF-based works [93–95] achieve keyless authentication but with the drawbacks of low accuracy or various design and implementation vulnerabilities. Due to the avalanche effect of hash operation, simpler structure of hashing the concatenated feature of user biometrics and the PUF responses suffers from even more server reliability problems than [95] since both biometrics and PUF reliability issues have to be overcome.

In this chapter, a “UDhashing” scheme is proposed to resolve this dilemma. User live biometrics (facial features are used as a case study example in this work) are extracted for user authentication and a PUF is used as the device signature provider. The proposed system adopts “fuse-on-device” and “match-on-server” schemes, i.e., the unification of user and device features into bio-code is performed at the end-device while the authentication of the bio-code is conducted by the server. In contrary to existing user-device authentication solutions [93, 94], both user and device identities are structurally and independently derived before they are fused. What is more, the proposed UDhashing scheme also eases the biometric feature extraction process as opposed to the “match-on-device” scenario [95]. The major contributions of this chapter are as follows:

1. The proposed “UDhashing” scheme binds a device to its user with the concept of random projection. A secret key is derived from its embedded PUF to transform its user’s facial features into a unique hash template (the bio-code) for authentication. Therefore, no secret has to be stored in the endpoint’s side and a successful authentication of the bio-code will require both a correct biometric input of a user to his/her registered device and a correct response to a query from that device.
2. An improved communication protocol is proposed to achieve mutual authentication between the endpoint and the server by using a zero-knowledge proof of knowledge of the endpoint’s secrets. Authentication from server to endpoint relies on the hashed PUF responses while the authentication from endpoint to server relies on the bio-codes. None of the hashed PUF responses or the bio-codes reveal the endpoint’s secrets but prove the authenticity of the corresponding prover. Since only the extracted feature in the hashed domain is transmitted, the system is resource efficient in protecting user privacy against lost/tampered/stolen biometric templates.

3. The proposed system achieves cancelable biometrics and system reconfigurability. The use of a strong PUF enables the one-time use of the authentication credentials. A hashed biometric pattern (the bio-code) leaks no information with this “one-time-use” policy and can be easily revoked, reissued or refreshed whenever necessary to prevent permanent compromise of the users’ biometrics or enable the system reconfiguration by applying a different challenge.

3.1 Related Works

3.1.1 Cancelable Biometrics based on Random Projection

Random projection is widely used for dimension reduction. The key idea stems from Johnson-Lindenstrauss (JL) Lemma [121], which can be stipulated as: Given $\varepsilon \in (0, 1)$, if m is a positive integer and $m \geq 4(\varepsilon^2/2 - \varepsilon^3/3)^{-1} \ln M$, then every high-dimensional dataset $X \in \mathbb{R}^n$ of M points can find a map $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ (with R being the projection matrix) in randomized polynomial time for any $u, v \in X$ such that the pair-wise distances are approximately preserved:

$$(1 - \varepsilon) \|u - v\|^2 \leq \|f(u) - f(v)\|^2 \leq (1 + \varepsilon) \|u - v\|^2 \quad (3.1)$$

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 [122].

Leveraging on this distance-preserving property, random projection has been introduced into cancelable biometrics [123–126], 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 [123, 126]:

- 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)¹ 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

¹The complexity of mGS is $O(nm^2)$ for a $n \times m$ matrix [127], which will not be a problem for IoT endpoint devices like laptops, tablets or smart phones as used in the target application scenario since those devices typically have more than several hundreds of GigaFlops of floating point performance.

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

orthogonal vectors than the classical GS. Its loss of orthogonality² depends only linearly on the condition number of original dataset and fails completely when r is singular [129, 128].

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

$$v = \langle R, y \rangle = R^T y \quad (3.2)$$

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.,

$$b_i = \begin{cases} 0 & \text{if } v_i \leq \tau \\ 1 & \text{if } v_i > \tau \end{cases} \quad (3.3)$$

The quantization threshold τ in Eq. (3.3) 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 enrolment phase.

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

$$\begin{aligned} \|v_1 - v_2\|^2 &= \langle v_1 - v_2, v_1 - v_2 \rangle \\ &= \langle v_1, v_1 \rangle - 2\langle v_1, v_2 \rangle + \langle v_2, v_2 \rangle \\ &= (R^T y_1)^T R^T y_1 - 2(R^T y_1)^T R^T y_2 + (R^T y_2)^T R^T y_2 \\ &= y_1^T R R^T y_1 - 2y_1^T R R^T y_2 + y_2^T R R^T y_2 \\ &\xrightarrow{R R^T = I} y_1^T y_1 - 2y_1^T y_2 + y_2^T y_2 = \|y_1 - y_2\|^2 \end{aligned} \quad (3.4)$$

The condition that $R R^T = I$ (I is an identity matrix) is required for Eq. (3.4) to be true. However, the random orthonormal matrix R (dimension: $n \times m$ ($n > m$)) obtained after mGS does

²Based 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 [128]. Therefore, the rounding error introduced by mGS will not be a problem for the proposed system using the monoPUF [24]. However, to better ensure the system performance, the loss of orthogonality of R can be calculated and compared with a threshold. Once a bad orthogonal matrix R is detected, the majority voting module would be triggered again to generate another PUF response r until a good orthogonal matrix R is obtained.

not strictly satisfy $RR^T = I$, but $RR^T = I + \delta_{i,j}$. Increasing m will provide better approximation of $RR^T \approx I$, and $\delta_{i,j}$ approaches 0 asymptotically when $m \rightarrow n$.

3.1.2 Constraints of Existing Biohashing Schemes

Despite the ease of implementation, the 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 [130]. 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. Applying a cancelable bio-code scheme can protect the biometric templates but shall not unduly degrade the authentication performance. Existing Biohashing schemes can achieve cancelable biometrics but the accurate authentication of both the user and the device can only be fulfilled with independent user-device authentication steps, making these Biohashing schemes inefficient in providing highly accurate user and device authentication as well as secure cancelable biometrics.

3.2 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. 3.1.2 in applying existing biohashing for this purpose.

3.2.1 Facial Feature Representation

In work [131], 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.

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

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 principal component analysis (PCA) and linear discriminant analysis (LDA) are not suitable due to their computational complexity. Instead, Gabor filters and the Discrete Cosine Transform (DCT) are used for feature extraction. Sparse representation based classification is applied for recognition.

Gabor Features

The 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 $\phi_{\mu,v}(x,y)$ is a Gaussian kernel function modulated by a sinusoidal plane wave whose center frequency and orientation are given by f_μ and θ_v , respectively [132], i.e.,

$$\phi_{\mu,v}(x,y) = \frac{f_\mu^2}{\pi k \eta} e^{-((f_\mu^2/k^2)x^2 + (f_\mu^2/\eta^2)y'^2)} e^{j2\pi f_\mu x'} \quad (3.5)$$

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

To obtain the Gabor features, a set of Gabor filters $\phi_{\mu,v}(x,y)$ is constructed, which typically has a size of five scales ($\mu = 0, 1, \dots, 4$) and eight orientations ($v = 0, 1, \dots, 7$) [132]. For a given face image $I(x,y)$, its Gabor feature corresponding to the parameter set (μ, v) can be calculated by

$$G_{\mu,v}(x,y) = I(x,y) * \phi_{\mu,v}(x,y) \quad (3.6)$$

where $G_{\mu,v}(x,y)$ 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.

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

3.2 Proposed Cancelable UDhashing Scheme

Fig. 3.1 (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.

the first few dominant AC 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. 3.1(a), the first N DCT coefficients of each block are extracted in a raster scan order, and then concatenated together as the local features [133].

3.2.2 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. A strong PUF with a practically inexhaustible number of CRPs is preferred for authentication. As each CRP will be used only once for each authentication session for the same user-device group, a replay attack is impossible and correlations between the system responses (i.e., the bio-code) cannot be exploited to predict the response.

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

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. 3.1, 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 enrolment phase, which will result in authentication failure. Therefore, an error correction code 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 a 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 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 be correctly guessed or predicted by unauthorized users. Secondly, strong PUFs can be vulnerable to modelling attacks [1, 76]. It is important to prevent the projected bio-codes from being attacked by the successful modelling of strong PUF. For these reasons, the prototype is designed with a diode-clamped inverter based strong PUF (monoPUF) [24] developed by our research group, 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 modelling attacks.

3.2.3 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

3.2 Proposed Cancelable UDhashing Scheme

degrade the feature quality, leading to unsatisfactory recognition results for later processing. As mentioned in Sec. 3.2.2, the high dimension of biometric feature and satisfaction of Eq. (3.1) require a proportionally large random sequence for the mGS algorithm. Even if strong PUF is adopted for the direct generation of the random matrix, for cost effectiveness, the concatenated features are segmented, and random projection is applied to each feature segment.

Fig. 3.1(b) shows an overview of the proposed UDhashing scheme. Instead of using a TPRN in bihashing, PUF responses are generated to build the projection space. Majority voting is applied before the mGS algorithm to enhance the reliability of the 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 resource-constrained end-devices. A simple procedure of using SRC for this tuple recognition is described as follows [124]:

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

$$\begin{aligned} D &= [D_1, D_2, \dots, D_L] \in \mathbb{R}^{m \times (sL)} \\ &= [x_{11}, \dots, x_{1s} | x_{21}, \dots, x_{2s} | \dots | x_{L1}, \dots, x_{Ls}] \end{aligned} \quad (3.7)$$

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.,

$$y = \sum_{i=1}^L \sum_{j=1}^s \alpha_{ij} x_{ij} = D\alpha \quad (3.8)$$

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 Eq. (3.8), 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 [134] and orthogonal least square [135]. Therefore, given an overcomplete dictionary $D \in \mathbb{R}^{m \times (sL)}$ for L distinct classes and a test sample $y \in \mathbb{R}^m$, the SRC algorithm can be described by: (1)

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

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 a 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) [136] 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.

3.2.4 UDhashing End-point Authentication Scheme

The parties involved are (at least) a user, (at least) a device, and a server. The proposed 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 performs the enrolment 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 strong PUF such as 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 re-route 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 in the device and the database on the server. 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

3.2 Proposed Cancelable UDhashing Scheme

are ways to design and implement the underlying algorithms such that their executions are side-channel aware [64].

An overview of the proposed system is shown in Fig. 3.1(c). The end-users are equipped with their end-devices, each of which is embedded with a CMOS camera and a built-in monoPUF. The flow of the lightweight mutual authentication protocol depicted in Fig. 3.2 is detailed as follows:

Fig. 3.2 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 $\tilde{T}_j^k$;

Enrolment Phase

The enrolment process is a one-time event that occurs in a secure environment at a trusted location. To initiate a legal enrolment 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

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

legitimate, and can be used to match any additional personal information such as password or signature. The server will acknowledge the successful enrolment 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 challenges 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 facial 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 the proposed UDhashing based scheme, and sent back to the server. The received bio-code for each (challenge, user, device) class is 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 enrolment, which can be achieved by an irreversible fuse or tamper-proof (public) storage of some sort [137]. Secondly, the stored face images will be deleted immediately upon successful enrolment. Thirdly, the enrolment process is conducted off line, with several face images to be taken from the user side. Fourthly, the bit length of the 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 [138], 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 [139] is applied to N and R_{uk}^i to generate a hash value A before N is sent to the server.

3.2 Proposed Cancelable UDhashing Scheme

- The server retrieves R_{uk}^i from its database, and calculates B using the same hash function after receiving N . B is then sent back to the endpoint. By this way, the server achieves zero-knowledge proof of the PUF response R_{uk}^i 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 is able to achieve zero-knowledge proof of the user biometrics and the device key by generating a “live bio-code” $\widetilde{T}_j^k$ according to the proposed UDhashing scheme and sending this bio-code 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. 3.2) can also be replaced by a PUF model with a strong PUF that supports model-based authentication[137, 140]. Utilizing a 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. 3.2. 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 the 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 enrolment 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 my previously reported scheme [131], adding new users will require all the existing accounts to be re-enrolled, which is impractical and infeasible in real applications.

3.3 Results and Discussions

3.3.1 Dataset and Experimental Setting

The experimental results are evaluated using the ORL [141] and Ext. Yale B [142] facial recognition databases. ORL is used as the genuine users while Ext. Yale B is used as the impostors. The 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 enrolment, while the remaining 4 images are used in the authentication phase for system performance test. The Ext. Yale B contains 38 subjects of 9 poses and 64 illumination conditions. For the 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 [24] 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.

3.3.2 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 pair-wise 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 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

 Fig. 3.3 Distribution of v and its corresponding mean μ for each $m(m')$.

to the distribution of v in Eq. (3.2) before the approximation of $RR^T \approx I$ and the resultant authentication performance are evaluated.

Fig. 3.3 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. 3.1.1, $RR^T \approx I$ 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 3.1 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 3.1 Mean μ_δ and standard deviation σ_δ 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. 3.4 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

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

Fig. 3.4 Normalized Hamming distance distribution for (a) (genuine user, genuine device) (b) (fake user, genuine device) (c) (genuine user, fake device) combinations with varying m .

corresponding authentication performances³ are shown in Table 3.2. 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. 3.4, 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. 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.

Table 3.2 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

3.3.3 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 3.3 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).

³The details about the authentication performance evaluation can be found in Sec. 3.3.4.

Table 3.3 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:

$$D = \frac{1}{N_{usr} \times N_t} \sum_{i=1}^{N_{usr}} \sum_{j=1}^{N_t} \frac{HD(T_i, T_j)}{l} \quad (3.9)$$

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. 3.5 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: $\mu = 0.497$, $\sigma = 0.015$.

Fig. 3.5 shows the normalized HD distribution for the five cases listed in Table 3.3. 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 bio-codes is $\sqrt{m}/(2m)$ (ideal $\sigma = 0.014$ for $m = 1300$) [34]. The distribution of all the impostor cases are clustered around

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

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.

3.3.4 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 3.4 shows that all five cases are correctly detected by the 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 3.4 Verification performance

Accuracy Case	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

3.3.5 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. 3.6(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 [24] for temperature variations from $-40^{\circ} \sim 90^{\circ}$ and supply voltage ranges from $0.9 \sim 1.3V$.

Fig. 3.6 (a) Authentication accuracy with unreliable device keys; (b) FAR versus average bit error rate of genuine bio-codes.

3.3.6 Comparison with Existing Work

Authentication schemes that unify both user and device characteristics are rare. Table 3.5 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 [95]. Note that though the work [95] and the proposed work can be classified as “UD-PUF” as introduced in Sec. 2.3.2, the two works actually have different research objectives. The work [95] targets **user access control** and data protection on device (“match-on-device”), while the proposed work targets **user-device access control** and data protection on server (“match-on-server”). In [95], the user-device key (obs_key) is generated by applying the hashed bio_key (extracted from the user biometrics) as the challenge to the PUF. Instead of using obs_key as the authentication credential in the server side, the obs_key is used as the key to the on-chip obfuscation logics. The obs_key has to be regenerated reliably each time to correctly unlock the device, which becomes a big concern of their system since biometrics are typically noisy and PUF response generation has the avalanche effect. Therefore, the work [95] focuses on evaluating the reliability issue of “bio_key”. Thanks to the proposed UDhashing scheme, the proposed work is free from this issue and focus more on evaluating the system discriminability performance.

Table 3.5 Comparison of user-device bipartite authentication schemes.

Scheme	Matching Site	System reconfigurability*	Cancelable biometrics	Mutual authenticity	Performance**
[93]	Not specified	No	No	No	Moderate***
[94]	Not specified	No	No	No	Moderate
[95]	Match-on-device	No	Yes	No	-
[131]	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.

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

Compared to the classical biometric authentication system, the cryptographic primitives used by the endpoint device are only a monoPUF [24], a TRNG [138] and a Spongent hash [139]. The 64-bit monoPUF consumes merely $3.85 \mu W$ power with an active silicon area of only $4719 nm^2$ [24]. The reported ASIC implementation of Spongent-128 consumes only 1060 gate equivalents and $2.2 \mu W$ [139], and TRNG consumes $214 nW$ at a throughput of $500 kbps$ with an area of $1400 \mu m^2$ [138]. Since these components are on standby until triggered by the authentication request, their actual power consumptions are lower. Moreover, the 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 [143]. The signal processing cost in UDhashing has also been largely reduced by the dimension reduction of raw biometric features.

3.4 Security Analysis

3.4.1 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 facial images are captured in real-time while the PUF response is only generated upon request. The cancelable bio-codes that indistinguishably unify 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, the proposed 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.

3.4.2 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 a certain percentage of bits in each bio-code of a genuine case to determine the corresponding false accept rate (FAR). Fig. 3.6(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 $f \leq 1 - 0.45 = 0.55$. 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

$$P = \sum_{i=\lceil Nf \rceil}^N \binom{N}{i} \left(\frac{1}{2}\right)^i \left(1 - \frac{1}{2}\right)^{N-i} = 2^{-N} \sum_{i=\lceil Nf \rceil}^N \binom{N}{i} \quad (3.10)$$

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 device⁴. 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. 3.3.5. To obtain a minimum above zero authentication accuracy of 3.13% based on the results of Fig. 3.6(a), the attacker needs to guess at least 60% of PUF response bits correctly. According to Eq. (4.25), 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.

3.4.3 Replay Attack

In the proposed system, the hashed credentials are transferred between an end-point and a verifier. For the endpoint to authenticate the server, the authentication credential is ‘ B ’ in Fig. 3.2. For the server to authenticate the endpoint, the authentication credential is the bio-code T_j^k shown in the same figure. Replay attack could be conducted by an attacker by eavesdropping the login credentials (B or the bio-codes), and then waiting for the same challenge from the verifier to replay the hijacked credentials. The attacker may collect many challenge- B pairs and challenge bio-code pairs (CBPs) of different genuine communication messages to increase his

⁴The case that the attacker obtains someone else’s facial images as well as the correct device can be viewed as a face spoofing attack, which can be mitigated using various countermeasures reported in the literature [144, 145]. Since this proposed work introduces a new general-purpose user-device bipartite authentication scheme, specialized research on detection of face spoofing attack is beyond the focus of this work. Nevertheless, existing countermeasures to face spoofing attack can be easily integrated into the proposed system. For example, it is possible to combine “face liveness evidence” information (expressions, speeches, depth information, etc.) into the biometric features before UDhashing. The accuracy and robustness of incorporating any such countermeasures can be explored as the future work.

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

chance of success. However, because of the introduction of random nonce N at the endpoint side, B is valid only for one time; As every unique challenge is also used only once, all the used challenge bio-code pairs collected by the attacker are useless for future replay. Therefore, the proposed system is robust against replay attack and the use of strong PUF ensures that enough CBPs can be generated for authentication throughout the lifetime of the device.

3.4.4 Man-in-the-middle Attack

A 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 the 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 $\widetilde{T}_j^k$ to masquerade as an authorized endpoint.

3.4.5 Modelling 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” $\widetilde{T}_j^k$ is directly accessible from the outputs of the module if the attacker has 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, a modelling-attack resilient strong PUF such as monoPUF is adopted in the proposed system. It was demonstrated that the modelling accuracy of advanced support vector machines [73] and reliability-based covariance matrix adaptation evolution strategy [76], even with a large training set of ten thousand CRPs,

on monoPUF has been reduced to around 50% [24]. If physical attack on PUF fails, black-box modelling 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.

3.5 Summary

This chapter presents a PUF-based scheme to achieve the unified authentication of both user and device. A novel UDhashing based mutual authentication scheme is proposed in this work. A modelling attack resilient strong PUF is utilized to enhance its security, and the authentication protocol has been optimized to minimize privacy leakage and conserve transmission bandwidth by only transmitting the user-device hash. Zero-knowledge proof was also achieved by using only the hashed secrets as the authentication credentials. Facial biometrics and PUF are integrated efficiently by UDhashing to provide cancelable biometrics along with bipartite authentication of both the device and its user. 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. The 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 modelling attacks. Note that the security of the proposed mutual authentication protocol can be further evaluated using automatic tools/programs such as ProVerif [146], CryptoVerif [147], Tamarin [148] and AVISPA [149]. 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 algorithms besides SRC.

Chapter 4

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

Image/Video security has been increasingly endangered by Artificial Intelligence (AI) as multiple forces are driving the latter to play a vital role in IoT applications. AI technologies can unlock IoT potential in multifold circumstances. For example, machine learning techniques can be used to automatically identify patterns and detect anomalies in the data that image sensors generate. AI with IoT can be used to analyze human behavior, and create truly grand smart home and smart shopping experiences. However, they can also be abused by malicious users to distort images or videos to mislead people or clinch wrongful convictions in a court of law. In late 2017, a software known as “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 [150]. This incident raises a red flag, given the prevalent use of surveillance footage to aid criminal investigation and civil litigation. As fraudsters can be adept at using AI, it is imperative to enhance digital (visual) evidence 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.

The review in Sec. 2.3.3 shows that both directions have been widely researched in their individual fields, however, to my 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 firstly proposed in [151], but there are deficiencies in that work. First of all, the work [151] is not resilient to geometric

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

transformations like rotation and scaling, which are common content-preserving manipulations in image processing. The experimental results of [151] were obtained from a very small and simple database, which is inadequate to demonstrate its robustness. Moreover, the method proposed in [151] 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.

This chapter presents a simple but effective way to achieve simultaneous tamper detection and source camera identification by using perceptual image hashing and PUF. 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 existing perceptual hashing methods that are only dependent on the data. The unique and innate device characteristics are 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 the tagged image and its origin is performed directly with the acquisition device without the need to store the CRPs of the PUF in a trusted server database.
4. An optimal selection of the 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.

4.1 Preliminaries

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

4.1.1 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. The SURF feature is highly invariant to scale, translation, lighting, contrast and rotation [152], 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 [153] for fast computation. The detection of scale and orientation invariant interest points is based on the determinant of a Hessian (DoH) matrix. A box filter is utilized to approximate the Gaussian second order derivatives, with which the approximated DoH can be calculated as [152]:

$$\det(H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \quad (4.1)$$

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 [152].

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 a $3 \times 3 \times 3$ 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 [154].

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

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

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:

$$D_i = [v_1, v_2, \dots, v_k, \dots, v_{16}] \in \mathbb{R}^{64} \quad (4.2)$$

where

$$v_k = [\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|], k = 1, 2, \dots, 16 \quad (4.3)$$

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

$$F = \{D_{1,s_1}, \dots, D_{i,s_i}, \dots, D_{m,s_m}\} \quad (4.4)$$

$$s_i = [\sigma, s\mathcal{L}, O, L, |det(H)|] \quad (4.5)$$

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 image feature matching [155] based on the computed descriptors. The MSAC-based technique (described in Sec. 4.1.2) is used to check the geometric consistency.

4.1.2 M-estimator Sample Consensus

The 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 [156].

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.

$$\text{rank}(CS) \stackrel{def}{=} |CS| \quad (4.6)$$

Let $\mathbf{D} = \{D_1, D_2, \dots, D_N\}$ 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

$$C_M(\mathbf{D}, \theta) = \sum_{i=1}^N \rho(D_i, M(\theta)) \quad (4.7)$$

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

$$\rho(D_i, M(\theta)) = \begin{cases} 0, & |e_M(D_i, \theta)| \leq \delta \\ 1, & \text{otherwise} \end{cases} \quad (4.8)$$

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

$$e_M(D_i, \theta) \stackrel{\text{def}}{=} \min_{D'_i \in M(\theta)} \text{dist}(D_i, D'_i) \quad (4.9)$$

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. (4.8) to

$$\rho(D_i, M(\theta)) = \begin{cases} e_M(D_i, \theta), & |e_M(D_i, \theta)| \leq \delta \\ \delta, & \text{otherwise} \end{cases} \quad (4.10)$$

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} [156] can be set to:

$$\tau_{stop} = \left\lceil \frac{\log \epsilon}{\log(1 - q)} \right\rceil \quad (4.11)$$

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 picked after τ_{stop} iterations is at most ϵ .

4.1.3 Physical Unclonable Function

In this work, a PUF is utilized to generate device-specific projection space for image features. Strong PUFs [1], such as arbiter PUF, have an exponential number of challenges relative to their number of bit-slices. While the practically inexhaustible number of challenges is good to

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

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 a PUF that can be intrinsically reused as another functional module in a computer system is particularly desirable to avoid the overhead of 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.

4.2 Proposed PUF-based Data-device Hash

To detect maliciously tampered, unscrupulously manipulated and fabricated images without restricting benign image processing and analysis, this work 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.

4.2.1 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 a 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.

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 the experiment), they 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:

$$F = \{D_{1,s_1}, \dots, D_{i,s_i}, \dots, D_{t,s_t}\} \in \mathbb{R}^{64 \times t}, t \leq T_f \quad (4.12)$$

where s_i is the descriptive information for each SURF feature point as denoted in Eq. (4.5).

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 [157]. The adjoint block-based DCT concatenated feature is proposed in [158] 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 tends 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.

$$F_{i,j} = [f_{2i-1,2j-1}, f_{2i-1,2j}, f_{2i,2j-1}, f_{2i,2j}] \in \mathbb{R}^{128} \quad (4.13)$$

$$i \in \{1, 2, \dots, M/2\}, j \in \{1, 2, \dots, N/2\}$$

where (i, j) is the row and column indexes of the *cblock*. $f_{2i-1,2j-1}$ denotes the *eblock* feature vector extracted from the *eblock* that resides in the $(2i - 1)^{th}$ row and $(2j - 1)^{th}$ column of the original image I .

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

CMOS Image Sensor Based PUF

To avoid sophisticated statistical image processing techniques, a CMOS image sensor based PUF [22] uses the innate fixed pattern noise (FPN) of an 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. 4.1, C_t is applied as a seed to 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 $2L_C L_R$ 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 cameras can be distinguished with high accuracy regardless of model type or brand.

4.2.2 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 from the concept of random projection (RP), specifically, 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 . RP is a widely used efficient dimension-reduction technique and the key idea stems from the *JL Lemma* [121] as introduced in Sec. 3.1.1. The random orthonormal matrix was used as it has been proven [122] to be qualified for the implementation of f in Eq. 3.1. The generation of such matrix usually requires orthogonalization methods like mGS algorithm to be applied, which

may not be implementation friendly to resource-constrained IoT devices. Alternatively, three other commonly used matrices Φ found in [159, 160] are applicable and they are listed below:

1. Independent and identically distributed samples from Gaussian distribution:

$$\Phi_{i,j} = \mathcal{N}(0, 1/m) \quad (4.14)$$

2. independent realizations of ± 1 Bernoulli random variables:

$$\Phi_{i,j} = \begin{cases} \frac{+1}{\sqrt{m}}, & \text{probability} = 0.5 \\ \frac{-1}{\sqrt{m}}, & \text{probability} = 0.5 \end{cases} \quad (4.15)$$

3. matrices whose entries satisfy:

$$\Phi_{i,j} = \begin{cases} +\sqrt{\frac{3}{m}}, & \text{probability} = \frac{1}{6} \\ 0, & \text{probability} = \frac{2}{3} \\ -\sqrt{\frac{3}{m}}, & \text{probability} = \frac{1}{6} \end{cases} \quad (4.16)$$

Since a PUF response is ideally a random vector of binary bits with uniform distribution, the projection space is proposed to be specified by a PUF response driven Bernoulli random matrix, which is easier to be implemented than the random orthonormal matrix as used in Sec. 3.1.1. The entries of the 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 .

$$P_i = \begin{cases} \frac{+1}{\sqrt{m}}, & R_i = 0 \\ \frac{-1}{\sqrt{m}}, & R_i = 1 \end{cases} \quad (4.17)$$

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. (4.12) and the adjoint DCT features $F_{i,j}$ of Eq. (4.13) into this PUF-specified Bernoulli random matrix, a data-device hash H can be generated as follows:

$$\begin{aligned} H &= [H^{SURF}, H^{aDCT}] \\ &= [P_{f1}^T D_{1,s_1}, \dots, P_{f1}^T D_{i,s_i}, \dots, P_{f1}^T D_{t,s_t}, \\ &\quad P_{f2}^T F_{1,1}, \dots, P_{f2}^T F_{M/2,N/2}], \quad t \leq T_f \end{aligned} \quad (4.18)$$

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

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 a lunchtime attack¹. 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.

Algorithm 1 Knuth Shuffle

1. **Input:** PUF response R_s , direct hash H of length l
 2. **for** $i = l$ **downto** 2 **do**
 3. $j = (\text{unbiasedRandIntGen}(R_s) \bmod (i - 1)) + 1$;
 4. **swap** $H(i)$ and $H(j)$
 5. **end for**
-

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.

4.2.3 Image Tampering Detection and Source identification

Fig. 4.1 shows the framework of the proposed PUF-based perceptual data-device hash in a digital forensic application scenario. As shown in Fig. 4.1, 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

¹A lunchtime attack refers to the idea that an adversary has access to an authenticated camera while the authorized user is out for lunch [161].

4.2 Proposed PUF-based Data-device Hash

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.

Fig. 4.1 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.

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, the 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

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

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. In the meantime, the challenge C_t stimulated PUF response R (divided into R_{f1} and R_{f2}) is applied to generate the Bernoulli random space P (P_{f1} and P_{f2} , respectively). By calculating $P_{f1}^T F_r$, 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.,

$$\begin{aligned} L_{I_o} &= matchedPtsOriginal.Location; \\ L_{I_r} &= matchedPtsReceived.Location; \\ \frac{\#(abs(L_{I_o} - L_{I_r}) \leq 5)}{\#(All_matchedPtsPairs)} &\geq 0.5 \end{aligned} \quad (4.19)$$

where $\#(A)$ is the cardinality of dataset A . In this case, I_t in Eq. 4.20 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:

$$I_t = \theta I_r = \begin{bmatrix} a_1 & a_2 & b_1 \\ a_3 & a_4 & b_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r \\ y_r \\ 1 \end{bmatrix} \quad (4.20)$$

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

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. 4.4.3). This situation may lead to limited or even no matched SURF points found. Under this circumstance,

Fig. 4.2 An example of wrong matching: (a) original image I_o ; (b) received image I_r ; (c) recovered image I_t .

even if the SURF feature detection and matching are conducted as normal, the recovered images are probably distorted as shown in Fig. 4.2. 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 a certain threshold (10% in the experiments) of the total image pixels, it is deemed as an image registration failure and the received image will be rejected immediately.

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 $cblock$ and compared with a tamper threshold τ_e to detect the tampering:

$$e_{i,j} = \sqrt{((H_t)_{i,j}^{aDCT} - (H_o)_{i,j}^{aDCT})^2} \quad (4.21)$$

$$i \in \{1, 2, \dots, M/2\}, j \in \{1, 2, \dots, N/2\}$$

The outcomes of these two stages are shown in Fig. 4.3.

Fig. 4.3 Received image authentication of proposed data-device hashing.

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

Adaptive threshold: An adaptive threshold is proposed to increase the robustness of the tampered image detection. The problem with using a fixed threshold for all the images in conventional perceptual image hashing [158, 104] 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. The linearly separable hyperplane is obtained 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,

$$\tau_e(I) = a \times \tilde{e}(I) + b \quad (4.22)$$

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 my 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.

4.3 Experiment Setting and Key Parameter Optimization

4.3.1 Dataset Preparation

A modified CASIA database² 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 [162], 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. 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.

²Download link: <https://iee-dataport.org/open-access/modified-casia>

4.3 Experiment Setting and Key Parameter Optimization

According to CASIA ITDE v2.0, the tampered images are generated using the 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, the CASIA dataset is enriched into 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. 4.4 shows D_{au} , $D_{tampered}$ and D_{au_cp} with 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. 4.4.

For the authorized cameras used in the experiment, their CRPs were simulated by eight PUF instances for a 128×128 CMOS image sensor array using a 180nm TSMC CMOS technology process design kit in the Cadence environment. The design of the CMOS image sensor PUF of [22] 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. 4.3.2. Monte-Carlo simulated results of the PUF designed for a 64×64 CMOS image sensor array were first validated by real silicon data measured from five 64×64 CMOS image sensor array PUF chips fabricated also in 180nm CMOS technology [22]. More instances of the 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

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

Fig. 4.4 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 is 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.

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 $\pm 11\%$ supply voltage variations.

4.3.2 Hash Dimension Selection

As introduced in Sec. 4.2.2, 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 the 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 $m = m_1 + m_2 = \text{round}(p \times n_1) + \text{round}(p \times n_2)$. According to JL Lemma, a lower ε means better preservation of Eq. (3.1), 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.

$$e_{max} = \max_{i \in [1, M/2], j \in [1, N/2]} e_{i,j} \quad (4.23)$$

4.3 Experiment Setting and Key Parameter Optimization

Fig. 4.5 (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 ~ 700.

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. 4.5 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 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. 4.5 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: $0 \sim 360^\circ$) 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. (3.1). The e_{max} distributions for $p = 0.1, 0.2$ and 0.3 have comparable IQRs and medians in the tampered case as well

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

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.

4.3.3 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 the original-received image pairs mentioned in Sec. 4.3.2. Fig. 4.6 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 the Bayesian Linear Classifier (BLC). The green line $y = 3.3 \times \tilde{e}(I) + 1030$ shown in Fig. 4.6 denotes the threshold found using the BLC-based classification method.

Fig. 4.6 Adaptive threshold determination.

Under rare circumstances where the tampered regions of the received image are exceptionally large or the received image is a completely different image to 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 a 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. 4.6) 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. 4.7. 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. (4.22) is set to:

$$\tau_e(I) = \begin{cases} 3.3 \times \tilde{e}(I) + 730, & \tilde{e}(I) \leq 500 \\ 2380, & \tilde{e}(I) > 500 \end{cases} \quad (4.24)$$

Fig. 4.7 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.

4.4 Performance and Discussion

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

4.4.1 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. 4.4 should be classified as authentic. The FRR of each content-preserving case is measured to evaluate the perceptual

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

robustness. Lower FRR indicates a better perceptual robustness. Table 4.1 shows that with $\tau_e(I)$ of Eq. (4.24) 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 would be needed to support the decision.

Table 4.1 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	$ra = 0 \sim 360^\circ$	0.0375
Scaling	$S_x = S_y = 0.5$	0.0958

4.4.2 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 not only have higher TDR, but also be 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 4.2 shows a high TDR of 95.42% over 240 test image pairs. With content-preserving manipulations on tampered images, the TDR is still $\geq 90\%$ though slightly lower than the “tampered only” case. The minor reduction in TDR is ascribed to the increased difficulty in the image registration phase as these manipulations may introduce more deviations in the hash distance comparison. The examples in Fig. 4.8 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.

Table 4.2 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	$ra = 0 \sim 360^\circ$	0.9250
+ Scaling	$S_x = S_y = 0.5$	0.9042

4.4.3 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 the 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 benchmark 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 PUF-based Data-Device Hash for Tampered Image Detection and Source Camera Identification

Fig. 4.8 Examples of tamper detection and location for all manipulations shown in Table 4.2. (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.

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. 4.9(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 evidenced by the average FAR of only 0.000208 in Fig. 4.9(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 gaps. For the use case of image forensics, they would be rejected by visual

Case 1: Change of challenge										
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

(a)

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

(b)

Fig. 4.9 Source camera identification performance.

inspection before being presented as 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 the 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.

4.4.4 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 the 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) [45] 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 4.3 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 satisfied by the CMOS image sensor PUF over the industrial grade of operating temperature variation and typical regulated supply with no more than 10% voltage variation. Moreover, the reliability of R can also be further enhanced by simple majority voting technique.

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

Table 4.3 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

4.4.5 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 a 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. 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.

A 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, the PUF

makes such an 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 re-fabricate 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 the proposed 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. Note 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 lunchtime attack as mentioned in the last two paragraphs of Sec. 4.2.2 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 lunchtime attacks impossible. As for the random guessing of R , Table 4.3 shows that on the premise of a 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:

$$P_r = \sum_{i=\lceil Nf \rceil}^N \binom{N}{i} \left(\frac{1}{2}\right)^i \left(\frac{1}{2}\right)^{N-i} = 2^{-N} \sum_{i=\lceil Nf \rceil}^N \binom{N}{i} \quad (4.25)$$

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

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 , and it is also nearly impossible to recover R_s . The authentication tries for the same image can also be limited to prevent brute force search of both R_s and R .

4.4.6 Comparison

In this section, the proposed work is compared with existing perceptual image hashing methods proposed 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 (≥ 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 4.4 for comparison. Table 4.4 compares the performance of the proposed work based on a single manipulation of the operations in Table 4.1 against the state-of-the-art perceptual image hashing methods. The experiment results show that the 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. Note that though the true positive rates for “Rotation” and “Scaling” are not as high as those of [105], the proposed method achieves a TDR of 0.9542 while the TDR of [105] 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.

Table 4.4 Comparison with existing works.

Scheme	Perceptual robustness (parameters)		Tamper Detection	Tamper Location	Device Authentication
	Rotation	Scaling			
TIFS2012 [103]	0.9075 ($2^\circ \sim 30^\circ$)	0.8818 (0.5 ~ 1.5)	Yes	Yes	No
TIFS2013 [163]	N/S (5°)	N/S (0.5, 1.5)	Yes	Yes	No
TIFS2015 [104]	0.8609 (25°)	0.8477 (0.5)	Yes	Yes	No
TIFS2016 [105]	0.9926 ($\pm 1^\circ \sim \pm 90^\circ$)	1 (0.5 ~ 2)	Yes	No	No
Springer2016 [96]	0.8002 ($\leq 5^\circ$)	1 (0.5 ~ 1.5)	Yes	Yes	No
AsianHOST2016 [151]	No	No	Yes	No	Yes
ISCAS2018 [158]	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. [103, 96]), the one with best performance is chosen for comparison.

4.5 Summary

This chapter presents a PUF-based scheme to achieve the unified authentication of both data and device for digital forensics. 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 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 the PUF response has been calculated to be almost impossible.

Chapter 5

Event-driven Dynamic Vision Sensor based Physical Unclonable Function for Camera Authentication in Reactive Monitoring System

As defined, PUF can be viewed as a mathematical mapping of challenges and responses. To my best knowledge, existing PUFs are only evoked by an input challenge, but are unresponsive to environmental changes¹. The lack of situation awareness has been a limiting factor for exploring PUF in attestation of perjury attributed to fraud and falsification of records. As video surveillance forms an indispensable part of IoT and becomes increasingly ubiquitous and pervasive, it is imperative to instil situation awareness into these monitoring systems for law enforcement so that their security module is able to response timely to suspicious/abnormal activities in its environment with a fidelity record of the incidents. The work in this chapter deals with the use of PUF for strengthening the witness testimonies of reactive video surveillance systems.

The most commonly used surveillance cameras are built from the frame-based image sensors, which generate continuous frames of images. Using frame-based imaging for video surveillance applications requires a tremendous amount of time/effort to sit through endless hours of video footage to look for evidence or clues of specific events, offences or exceptions, which is not only inefficient but also infringes on privacy for applications like assisted living [164]. Reactive monitoring of such systems can only be achieved by analyzing the video streams using post-processing techniques. Besides, additional sensors like radar or laser range finders

¹Temperature/voltage/... variations will be reflected partially in some bit flips of unreliable PUF response but they do not trigger PUF operation.

Event-driven Dynamic Vision Sensor based Physical Unclonable Function for Camera Authentication in Reactive Monitoring System

are usually needed in order to reduce redundant recording. These mechanisms inevitably introduce system delay. Moreover, frame-based image sensors are often limited by their burst mode frame rate and frame buffer capacity for very high-speed applications like traffic monitoring and crime tracking (e.g., mid-flight trajectory of bullet) [165]. These problems can be eased by using emerging Dynamic Vision Sensor (DVS) [166] and the augmented Dynamic and Active Pixel Vision Sensor (DAVIS) [167]. DVS pixels respond asynchronously to relevant changes in intensity and output them in sparse address-event representation (AER). The reduction of data redundancy by event-triggered focal-plane analog processing makes DVS more suited to reactive surveillance systems than conventional frame-based image sensor.

Due to rampant attacks on video cameras and proliferated entry points in visual sensor networks, using reactive monitoring for law enforcement requires the guarantee not only of the integrity of the collected evidence but also of the authenticity of the sensing device. In order that the footage captured by the camera can be used to hold the offender accountable for his misdeed, the trustworthiness of the DVS camera must be established and indelibly imprinted into the video footage. The PRNU-based individual camera authentication methods reviewed in Sec. 2.3.3 targeting frame-based image sensors are vulnerable and their device source detection accuracy is highly constrained by the image fidelity and quality. Those methods are not feasible for DVS since DVS outputs discrete temporal-spatial information instead of perceptual intensity values directly perceivable by human eyes. A more practical solution is to derive a unique hardware security module directly from the sensor node. A PUF that is driven by DVS event data is a fresh angle to this problem. As reviewed in Sec. 2.3.1, PUF-based authentication of frame-based image sensor has been proposed in [22]. The image sensor signature is extracted from the FPN of the reset image by bypassing the correlation double sampling (CDS) of the active pixel sensor (APS). Unfortunately, the CDS bypass is not applicable to DVS due to their completely different sensing and reset mechanism. Besides, the frame-based image sensor PUF is a passive sensor ID for authenticating the sensor node. It does not offer the “activeness” per se that is desirable for irrefutable testimony evident against record fabrication. As no exposure can be made during PUF operation, the response to the input challenge has nothing directly related to the image data or video footage acquired by the sensor. To provide non-repudiable evidence of the occurrence of any incident captured by a trusted camera, such a truly coherent link is strongly desirable for reactive surveillance system but is currently unachievable.

This chapter presents a novel event-driven PUF to enhance the security and reduce the redundancy of reactive surveillance system. The main idea is to build a PUF out of a DVS image sensor array with the generated PUF responses dependent on the footage. The response of an event stream package can only be triggered automatically by its first address event instead of an external challenge. The following original contributions will be presented in this chapter.

1. The proposed PUF is event driven and designed as a system instead of a standalone security primitive. New sensor architecture is designed with negligible pixel modifications and enhanced security. The entropy source is theoretically analyzed to be sensitive to process variations. A non-intrusive readout scheme is devised to operate in tandem with the continuous-timed and self-timed reset switches of DVS pixels. Ed-PUF can actively respond to the motion or activities in the scene, which greatly improves the responsiveness, reduces the data redundancy and cuts down the cost of auxiliary non-vision based sensors for event detection.
2. The proposed system is able to check the data integrity and authenticate the source camera concurrently. This is achieved by utilizing the Ed-PUF response as the key to hash the event stream based on the Hash-based Message Authentication Code (HMAC) technique. Secure syndrome construction based reverse fuzzy extractor technique is adopted to mitigate the PUF reliability problem. Such PUF-based HMAC scheme ensures secure transmission over the communication channel as the raw PUF responses are not transmitted for authentication.
3. A dedicated enrolment and authentication protocol is proposed for the Ed-PUF system. Logic obfuscation is added to the DVS sensor to prevent malicious users who has access to the camera from wiretapping by severing or splicing into the connections between the sensor chip and the processor to collect raw PUF responses. The unlock key is derived from the Ed-PUF response and changes for every key-request session.
4. The system performance is comprehensively evaluated by more accurate post-layout raw baseline PUF data and event streams acquired by a DVS camera in real scene. An upper bound of the collision probability is also derived and validated by calculation from the experimental data. The implementation overheads are reasonably small for a trusted video surveillance system.

5.1 CMOS Vision Sensor Preliminaries

5.1.1 Frame-based Active Pixel Sensor

The active pixel sensor (APS) is a widespread frame-based image sensor composed of an array of pixels where each pixel has a photodetector and an active amplifier. Fig. 5.1 shows the two most common pixel structures, which are a three-transistor (3T) and four-transistor (4T) APS. The photodiode senses the light and converts its intensity into electrical current (i.e., photocurrent). Both the 3T- and 4T-APS have a reset switch (*rst*), a source follower (*sf*) and an access transistor to reset, buffer and read out the voltage of the photodetector node, which

Event-driven Dynamic Vision Sensor based Physical Unclonable Function for Camera Authentication in Reactive Monitoring System

is linearly proportional to the photocurrent during the integration stage. The 4T-APS has an additional transfer gate (tg) to improve sensitivity and reduce dark current [6].

Frame-based APS operates by alternating between reset and integration modes, and CDS performs noise cancellation by subtracting the pixel reset signal from the sensed signal values [22]. The integration voltage is read out as an indicator of the light intensity sensed by the pixel. There are two main drawbacks of this pixel structure. Firstly, the integration voltage only respond to the average photocurrent (within an exposure time), which means the final obtained information is only related to an average intensity while the transient change in the scene is not well-reflected. Secondly, frame-based data acquisition leads to a large amount of redundancy in data readout and processing. In order to extract useful information, consecutive data frames have to be generated and transmitted with huge amount of computation dedicated to the subsequent different complex image processing algorithms. Increasing frame rate helps to improve the transient responsivity but generates even more redundant data and imposes greater computational limitations.

Fig. 5.1 Schematic of typical (a) 3T-APS and (b) 4T-APS for the frame-based image sensor.

5.1.2 Event-based Dynamic Vision Sensor

DVS is an event-based image sensor, which was first reported in [166] as an excellent supplement to the widely existed frame-based APS. DVS adopts a totally different design strategy from APS. Each pixel in DVS monitors the local light intensity change continuously and generates asynchronous spikes externally when the detected intensity exceeds some pre-defined threshold. While APS has a very simple 3T or 4T pixel structure, the DVS pixel is much more complex. The block diagram *Pixel_o* of a typical DVS pixel can be found in Fig. 5.2(a), which is composed of two main parts, *Pixel_Core* and *Pixel_AER*. The transistor-level schematic of *Pixel_Core* is shown in Fig. 5.2(b). The photodiode and transistors $M_1 \sim M_4$ form a logarithmic receptor, which converts the sensed photocurrent into a logarithmic voltage V_p by operating M_1 at subthreshold region. M_5 and M_6 form a source follower to buffer V_p . The transistors in the shaded area form a capacitive-feedback transimpedance amplifier (CFTA), which provides a large voltage gain C_1/C_2 (≈ 20) to increase the temporal contrast sensitivity. The output V_{diff}

Fig. 5.2 (a) *Pixel_o*: Block diagram of a typical DVS pixel; (b) *Pixel_Core*: Transistor-level pixel schematic of a DVS; (c) *Pixel_AER*: Asynchronous event representation generator logic of the pixel.

of the amplifier is then fed into two subsequent comparators (M_9 - M_{10} and M_{11} - M_{12}). Once V_{diff} is increased or decreased to some pre-defined threshold (V_{ON} or V_{OFF}), a corresponding “on” ($0 \rightarrow 1$) or “off” ($1 \rightarrow 0$) signal will be sent out. DVS adopts a self-timed asynchronous reset mechanism. Once the detected event has been completely processed, an internal reset signal (labelled in yellow in Fig. 5.2) is automatically generated to turn on the reset switch M_r in Fig. 5.2(b). As a result, V_{diff} will return to its reset level $\overline{V_{diff}}$ to prepare for another event detection. The operating principle of DVS is shown in Fig. 5.3(a).

Fig. 5.3 (a) Operating principle of DVS. (b) 4-phase handshaking protocol to process the fired pixel [6].

DVS adopts Address Event Representation (AER) logic for data output. The detailed AER logic of a pixel *Pixel_AER* is shown in Fig. 5.2(c). Once a pixel is fired, i.e., when an “on” or “off” event is detected, a 4-phase handshaking protocol is used to process the fired pixel. As illustrated in Fig. 5.3(b), the fired pixel first issues a row request (*RR*) signal. If the row of the fired pixel is selected, a row acknowledge (*RA*) signal will be granted for the fired pixel to send its column request (*CR*: either *CRON* or *CROFF*) signal. Upon receiving a column acknowledge

(CA) signal, the fired pixel will be processed. Consequently, a single-bit event polarity (to indicate either an “on” or “off” event is detected) and a multiple-bit row/column address will be reported to the off-chip processor². The event will be time-stamped automatically once it is received by the processor.

5.2 Proposed Event-driven PUF

5.2.1 New DVS Pixel Circuit

As mentioned in Sec. 5.1.2, a DVS pixel will be reset automatically after every event generation cycle. Ideally, the reset level $\overline{V_{diff}}$ should be equal among identically designed pixels, but in reality, $\overline{V_{diff}}$ will vary due to the inevitable process variations. Though imperfect for normal DVS operation, $\overline{V_{diff}}$ becomes an exploitable entropy source for designing a unique PUF out of DVS. However, due to completely different sensor architecture, the simple way of “resetting the entire pixel array and bypassing CDS to read out the reset voltage for PUF response generation”, which works for the frame-based APS image sensor based PUF in [22], is infeasible to build a functional PUF from DVS. The “CDS bypass” technique is feasible in [22] because the entropy source of the PUF is the pixel reset value (obtained by turning on the reset switch in Fig. 5.1), which is the input to the CDS. Such global static reset state is absent in DVS. The entropy source of DVS $\overline{V_{diff}}$ is the reset value of the CFTA. Although the CFTA (pixel-wise) of DVS has similar structure as the CDS (column-wise) in the conventional image sensor, bypassing the CFTA will directly lose the entropy source. Besides, reset and integration are mutually exclusive in frame-based imager. This, on one hand, provides free isolation between PUF mode and imaging mode, but on the other hand makes spontaneous input image driven PUF infeasible as the two functions cannot be run concurrently. On the contrary, individual pixels of DVS can be reset at any time during exposure except that the reset cannot be controlled by the user but the activity in the scene. The entropy source that can be extracted for PUF response generation is $\overline{V_{diff}}$. The voltage at this internal node responds to intensity change continuously and reset momentarily upon reaching the threshold to output the address event. There is no access path to read out this node voltage in the original pixel circuit of DVS; Due to the self-timed reset mechanism, the pixel is reset asynchronously upon completion of an address event, and the firing of each pixel is uncontrollable. This makes the extraction of $\overline{V_{diff}}$

²Though not explicitly indicated in DVS products, a promising processor candidate is the DynapCNN, where Dynap and CNN are the acronyms for “Dynamic neuromorphic asynchronous processor” and “Convolutional Neural Network”, respectively. DynapCNN [168] provides dedicated interface to DVSs and takes the event stream from most of the advanced DVSs as direct input. It is scalable, fully-configurable, saves 100-1000 times more power than the state of the art, and enables ultra-low-latency of below 5ms. Therefore, DynapCNN is most suited for ultra-low power, real-time and always on applications.

Fig. 5.4 New pixel design for the proposed Ed-PUF: (a) Transistor-level schematic; (b) Block diagram; (c) Simplified key component circuit for PUF mode of operation (the three added transistors are drawn in red).

difficult even if there is an accessing path. Since reset and exposure modes are not isolated, normal DVS imaging operation and PUF operation will interfere with each other, making the scheme in [22] inapplicable.

To tackle the above problems, we propose to create a non-intrusive readout path for $\overline{V_{diff}}$ with the help of the existing continuous-timed reset switch M_{er} . As shown in Fig. 5.2(b), M_{er} is another reset switch parallel to M_r . Unlike M_r , M_{er} is externally accessible and is originally built into the pixel to facilitate efficient processing of pixels in the regions of interest by keeping those rows of uninterested pixels in the reset mode. Other than this, M_{er} is not used for normal address event detection. As shown in Fig. 5.4(a), with the help of M_{er} , only three transistors (M_{13} - M_{15}) are added into the original DVS pixel in order to make PUF response generation feasible. A source follower consisting of two transistors, M_{13} and M_{14} , is used to isolate and drive the large capacitive load of the readout bus. M_{15} is an access transistor controlled by the external reset signal $\overline{rst}$. Only when the external reset switch M_{er} is turned on (for PUF operation) will the reset level $\overline{V_{diff}}$ be accessed. It also avoids collisions by allowing only one single selected row ($\overline{rst} = 0$) to report their $\overline{V_{diff}}$ to the shared column buses. The new DVS pixel diagram is shown in Fig. 5.4(b), where an inverter is added in each row to create an access control signal to M_{15} of all pixels in the row. The new DVS pixel circuit provides a non-intrusive readout path of $\overline{V_{diff}}$ to the analog output port ao in each column of the DVS pixel array. The additional reference voltage V_{ref2} can be easily adapted from existing configurable Bias Generator of DVS with only $360 \times 22 \mu\text{m}^2$ added per die [169].

5.2.2 Process Variation Analysis

As described in Sec. 5.1.2, the subcircuit in the shaded area of Fig. 5.2(b) is a transimpedance amplifier with a gain of C_1/C_2 . Small perturbation of the local intensity change will be amplified and reflected in V_{diff} . To avoid uncontrollable influence caused by the light intensity, the pixel has to be operated in fully reset condition to obtain a clean $\overline{V_{diff}}$ for PUF operation. In this premise, the capacitances C_1 and C_2 can be equivalently viewed as “open circuit”. Therefore, the key components that operate in PUF mode can be extracted and simplified to Fig. 5.4(c) for analysis.

The drain currents of the transistors, M_7 , M_8 , M_{13} and M_{14} , that operates in the saturation region can be approximated by:

$$I_D = \frac{W}{2L} \mu C_{ox} (V_{gs} - V_{th})^2 \quad (5.1)$$

where W and L are the gate width and length, respectively, μ is the charge-carrier mobility, C_{ox} is the gate oxide capacitance per unit area, V_{gs} is the gate-source voltage and V_{th} is the transistor threshold voltage.

Since M_7 and M_8 have the same drain current, we have $I_{D7} = I_{D8}$. Therefore:

$$\begin{aligned} & \frac{W_7}{2L_7} \mu_7 C_{ox,7} (V_{dd} - \overline{V_{diff}} - |V_{th,7}|)^2 \\ &= \frac{W_8}{2L_8} \mu_8 C_{ox,8} (V_{ref1} - V_{th,8})^2 \end{aligned} \quad (5.2)$$

Let $\beta = \sqrt{\mu C_{ox} W/L}$, then

$$\overline{V_{diff}} = V_{dd} - |V_{th,7}| - \frac{\beta_8}{\beta_7} (V_{ref1} - V_{th,8}) \quad (5.3)$$

V_{sf} is the output voltage of the source follower. With no loading, the drain current of M_{13} and M_{14} are the same. Similarly, we can express V_{sf} as:

$$V_{sf} = \overline{V_{diff}} - V_{th,13} - \frac{\beta_{14}}{\beta_{13}} (V_{ref2} - V_{th,14}) \quad (5.4)$$

Taking Eq. (5.3) and Eq. (5.4) collectively, we have

$$\begin{aligned} V_{sf} = & V_{dd} - |V_{th,7}| - V_{th,13} \\ & - \frac{\beta_8}{\beta_7} (V_{ref1} - V_{th,8}) - \frac{\beta_{14}}{\beta_{13}} (V_{ref2} - V_{th,14}) \end{aligned} \quad (5.5)$$

According to Eq. 5.5, V_{sf} depends on three external voltages (supply voltage V_{dd} and reference voltages, V_{ref1} and V_{ref2}) and transistor parameters including the threshold voltage, geometry and capacitance density of M_7 , M_8 , M_{13} and M_{14} . Since V_{dd} , V_{ref1} and V_{ref2} are shared among all the cells in a die, they have to be held constant and throughout the sensor array to avoid deteriorating the performance and quality of the original DVS functionality even without the PUF operation. This is typically assured by robust voltage reference [170] and careful layout techniques [171] to largely suppress noise, crosstalk and temperature induced variations. Therefore, the variations of buffered $\overline{V_{diff}}$, i.e., V_{sf} from pixel to pixel and across pixels from different dies of the wafer, are contributed mainly by the device manufacturing process variations.

5.2.3 Sensor Architecture

Based on the theoretical analysis of the entropy source of each pixel, and the proposed new pixel circuit for its accessibility, the DVS sensor architecture for the proposed Ed-PUF is depicted in Fig. 5.5. An $M \times N$ pixel array is formed based on the new pixel circuit shown in Fig. 5.4(b). The *Row/Column Handshaking Logic & Row/Column Encoder* blocks are used to execute the 4-phase asynchronous handshaking protocol and encode the fired pixel address during the event firing stage. Besides, the *Row/Column Arbiter Tree* modules are included to avoid collisions when multiple pixels in different rows and columns are firing at the same time. This way the original DVS outputs, i.e., the event polarity and event address (row & column address), are kept without modification.

Event Triggered PUF Operation

The PUF mode is triggered by events. As shown in the *Event_Gen* block of Fig. 5.5(a), an AND gate and a switch are added per column. The column request signal of a pixel is kept high in normal circumstance. Only when the corresponding row is selected by RA ($RA = 1$) will it turn to '0'. When a fired pixel is acknowledged by CA to output its address event, an *event_gen* pulse will be sent out to indicate the arrival of an event. The *event_gen* pulse is used as the clock of the Mod- Q counter, which will generate a trigger signal after counting Q arriving events. The trigger signal will activate the *Sample & Latch* module to capture the current event address. Meantime, a C -seeded LFSR will be triggered to generate a random state. C is an external challenge, which can be stored in the non-volatile memory (NVM) of the DVS sensor

Fig. 5.5 Block diagram of the sensor architecture of the proposed DVS-based Ed-PUF system. (a) The PUF-embedded sensor chip with DVS event generation and PUF operation components. *Capacitance Change Detector* is used to detect severing or splicing of PUF response transmission line. The last modulo reduction block in *Challenge_Gen* is not required if the row and column number of the array is a power-of-two integer like 128, 256. (b) Key-gates and activation signals for logic obfuscation; (c) Load capacitance perturbation detector module; (d) Overview of the communications between the proposed sensor chip of (a) and an off-chip processor embedded within the DVS camera.

and updated by the server over a certain period. The LFSR state and the latched event address are then “XORed” to generate the internal challenge C'^3 .

C' has the same bit length as the event address and is the real challenge that is used to stimulate the PUF. For an $M \times N$ sensor array, the first $\lceil \log_2 M \rceil$ bits of C' (C'_1) will be applied to the row decoder to select one row to reset. M_{er} and M_{15} of all pixels in the entire selected row will be turned on by $\overline{rst}$ to enable their respective $\overline{V_{diff}}$ value to be read out from the ao port in each column. The DVS function of this particular row is turned off during this time. Only event occurs in this row at this time will not be detected. All the other rows will operate as per normal and can concurrently fire events without being interrupted. Pairs of analog voltages from adjacent ao ports are applied to the column sense amplifiers to generate $N/2$ digital bits. $\lceil \log_2 N - 1 \rceil$ bits of the remaining bits of C' (C'_2) are applied to the multiplexer to select one of these digital bits as the final Ed-PUF response bit PUF_out for this fired event. The excess bit of C' is discarded. As such, every Q events will generate one Ed-PUF response bit. To generate an l -bit Ed-PUF response, $Q \times l$ events are needed.

Key-based Response Obfuscation

For added security against compromising the connections between the sensor chip and the processor (for example, wiretapping the connection by severing or splicing the wire), the sensor chip is locked in abnormal conditions. This can be accomplished by using key-based logic locking [172, 173] for response obfuscation⁴. As shown in Fig. 5.5(b), simple obfuscation gates, like a multiplexer for keeping/changing an internal connection from x or y to z , and XOR/XNOR for keeping/inverting the logic state of a signal x , can be inserted randomly between registers of the *Serial to Parallel Converter* circuits of the sensor chip. Instead of using a static key bit k_i for the control signal s_i of each obfuscation gate, we propose to control the obfuscation by mixing the key bit k_i with a random and refreshable bit r_i by a simple invertible function like $s_i = k_i \oplus r_i$, where $\oplus$ is an XOR operator.

The obfuscation is triggered when a wire severing or splicing attack is detected, which can be achieved by monitoring the load capacitance of the response transmission line with an on-chip relaxation oscillator (RO) [174, 175]. As depicted in Fig. 5.5(c), the RO will generate a non-sinusoidal repetitive output signal, whose frequency is determined by the capacitance being monitored. A counter is used to register the number of oscillatory cycles within a given

³ To improve the randomness of LFSR output, a larger LFSR with maximal-length polynomials is implemented and only the lower order bits are used to XOR with the latched event address. In the experiment, $\text{length}(C) = 24$ while $\text{length}(C') = 16$.

⁴ Such a protection is needed since off-chip processor is widely used to process the image data in current cameras. For future system-on-chip implementation, the connection between the sensor and the processor can be protected by replacing the logic lock with cryptographically secure active shield.

Event-driven Dynamic Vision Sensor based Physical Unclonable Function for Camera Authentication in Reactive Monitoring System

period. Perturbation of the load capacitance will cause a deviation of the oscillation frequency from the standard load. An obfuscation trigger signal t_{obf} will be sent out if the counter value falls outside the range for nominal frequency count f_{ref} . Once t_{obf} is on, the refresh of random bits r_i will be activated. The first event package detected after the assertion of t_{obf} is denoted as e_{obf} . r_i is refreshed by generating it from the l -bit PUF response using e_{obf} . The corresponding helper data h_{obf} of r_i will be calculated as well. Note that under this situation, only e_{obf} and h_{obf} will be transmitted (while in other situations, e_{obf} , h_{obf} and r_i will all be sent to the processor). r_i are registered internally until the arrival of the next t_{obf} , which will trigger the generation of a new r_i from a different batch of events e_{obf} . A correct input key $k_i = s_i \oplus r_i \forall i$ is required to recover the original connection or logic value. A wrong input key will keep the sensor in locked state by either outputting constant or incorrect PUF responses depending on the obfuscation logic and strategy.

Further Notes

Several important features are observed in the proposed Ed-PUF design. First and foremost, the DVS event rate is very high (in sub- μs per event). A small movement in the scene can generate more than enough event data to produce a cryptographically secure length of PUF responses (eg., 256 bits) for device authentication or event attestation. Besides, it is not necessary to trigger the PUF operation for every detected event. The Mod- Q counter can be adjusted to determine the sampling rate. As an example, if $Q = 32$, $l = 511$ and the event is assumed to be generated every $5 \mu s$ on average, the event stream period is calculated to be less than $0.08 s$. This can be easily satisfied in real situation. Secondly, the triggered PUF operation has its own separate processing bus without interrupting the 4-phase handshaking protocol. Therefore, the time required to report a fired pixel remains the same in the proposed scheme. Since DVS generates event asynchronously, only one selected row is inhibited from firing while events in all the other rows can continue to fire uninterrupted. Different from frame-based image sensor, DVS only detects/acquires address events when there are intensity changes. Therefore, the loss of address event from one row for a brief time does not impact event visualization, as will be demonstrated in the experimental results in Sec. 5.3.3.

5.2.4 Secure Transmission

The DVS-based Ed-PUF is still a weak PUF by definition. Therefore, it is important to protect the PUF interface and avoid sending raw PUF response for authentication. As explained in Sec. 5.2.3, once a movement is detected in the scene, the DVS sensor will report the detected events as well as the associated Ed-PUF response to its off-chip processor, where the

received event data will be tagged with the temporal information automatically. To ensure a secure transmission between the DVS camera and the server, the generated data will be further processed using a Hash-based Message Authentication code (HMAC) before its transmission. HMAC involves a cryptographic hash function H (of any type) and a secret key k . The hash tag generated using a HMAC for a message m can be expressed as [176]:

$$\begin{aligned} tag &= HMAC(k, m) = H(k_{out} \parallel H(k_{in} \parallel m)) \\ &= H((k' \oplus opad) \parallel H((k' \oplus ipad) \parallel m)) \end{aligned} \quad (5.6)$$

where $\parallel$ denotes concatenation. $opad$ and $ipad$ are block-sized outer and inner paddings, respectively. Specifically, $opad$ consists of repeated bytes valued 0x5c and $ipad$ consists of repeated bytes valued 0x36. k' is a block-sized key derived from the secret key k by hashing down then right padding with zeros to the block size, i.e.,

$$k' = \begin{cases} H(k) & \text{length}(k) > \text{block_size} \\ k & \text{otherwise} \end{cases} \quad (5.7)$$

Fig. 5.6 (a) The generation of an l -bit Ed-PUF response; (b) PUF-based HMAC for tag generation with syndrome construction based reverse fuzzy extractor incorporated to deal with the PUF reliability problem.

In the proposed case, the secret key in the HMAC scheme is the Ed-PUF response. As shown in Fig. 5.6, every event contains a 1-bit polarity, a S -bit ($S = \lceil \log_2 M \rceil + \lceil \log_2 N \rceil$) address and a T -bit timestamp information, and every $Q \times l$ events forms a piece of message. Assume that the response triggered by a piece of message m_1 is R_1 . R_1 is used as a key to generate the HMAC tag tag_1 for m_1 . SHA3 is used as the hash function for HMAC as it is immune to length-extension attacks. Note that the secret key has to be perfectly error-free in order to obtain the same HMAC output but 100% reliable PUF response is unattainable. To solve this paradox with low computation and hardware cost at the end point device, syndrome

Event-driven Dynamic Vision Sensor based Physical Unclonable Function for Camera Authentication in Reactive Monitoring System

construction based reverse fuzzy extractor [64] is adopted. To be more specific, a helper data h_1 can be calculated from the potentially noisy response generated by the PUF device for authentication by executing $R_1 \cdot H^T$, where H^T is the transpose of the parity check matrix of a linear block code like the Bose-Chaudhuri-Hocquenghem (BCH)⁵ code. The helper data, together with the event stream and the corresponding tags are packed together and sent to the server. The server can then recover the same noisy response R_1 as generated in the device side with the help of h_1 and the enrolled R_1 stored in its database by performing $R_1 = (R_1)_{enrolled} - e$, where e is obtained by calculating $h_1 - (R_1)_{enrolled} \cdot H^T = e \cdot H^T$. Such reverse fuzzy extractor scheme has been proven to possess outsider chosen perturbation security [64, 65], i.e., the system remains secure even if the attacker can collect multiple helper data from various noisy version of the same PUF response. The challenges raised in [66] are not applicable to our system as the proposed Ed-PUF is evaluated to have balanced 0 and 1 distributions and a trivial disparity of $\leq 0.16\%$ between the bit error rates (BERs) of 0 and 1.

5.2.5 Authentication Protocol

The proposed scheme allows the server to authenticate the sensor as well as validate the data integrity. Enrolment is required before the DVS sensor is deployed in the field. Any abnormal capacitance perturbations detected on the response transmission line between the sensor chip and the processor would lock the camera. As long as the camera is unlocked by a correct key, the events detected and transmitted by the camera can be authenticated. The server is assumed to have authorized access to the enrolled PUF database. The enrolment phase is conducted in a safe environment. Upon camera deployment, it is assumed that the attacker can reach out for the DVS camera or compromise the transmission channel.

- **Enrolment Phase:** The enrolment for the Ed-PUF is difficult if all the (event, challenge C , response) pairs need to be collected. Instead, C'_1 (as shown in Fig. 5.5) will be controlled to generate all the possible row addresses to iteratively reset the rows. The digital output $SADO$ of the sense amplifier, after reading in the $\overline{V_{diff}}$ values of the reset row from adjacent ao ports, will be exported and securely saved. This archive of “ $C'_1 - SADO$ ” mapping is actually the challenge-response-pair (CRP) mapping of a baseline PUF⁶ and it will be safely kept in the server as the PUF database for authentication. Besides, f_{ref} in Fig. 5.5(c) needs to be measured and stored in the NVM of the sensor chip. After the enrolment has been completed, all the measurement pads (C'_1 , $SADO$ and NVM read/write pins) will be permanently disabled.

⁵In my experiments, [511, 238, 37] BCH code is used to achieve a failure rate of less than 10^{-6} .

⁶In this work, the PUF whose response generation depends solely on a challenge is defined as a “baseline PUF”. The PUFs tested in [177, 178] are all “baseline PUFs”.

Fig. 5.7 Authentication protocol of the proposed system. $\hat{R}$ is the noisy version of the correct R , which can be retrieved from the enrolled PUF database DB with the help of helper data h . tag_re denotes the recalculated tag. The command to “refresh C ” is denoted by dashed arrow since it may or may not be given depending on the rule set in the server.

- Authentication Phase:** The camera is able to report event packages as well as their tags to the server. During the transmission process, the data may be modified or replaced by an attacker. The server can authenticate the data before making important decisions based on the received event data. To do this, the server needs to first retrieve the internal challenge C' from the received events. The first event in an event package is the event that triggers the PUF response generation by definition. Therefore, the internal challenge C' can be retrieved by knowing the stored external challenge C (or the refreshed C) and the received event address. After retrieving C' (which can be divided into C'_1 and C'_2), the server is able to locate $SADO$ by looking up in the “ $C'_1 - SADO$ ” database, and find the corresponding multiplexer output (the first bit of the PUF response) according to C'_2 . Similarly, every C' can be retrieved for every Q events in the event package. All the l -bit Ed-PUF response R can be successively retrieved in the same manner. With this R and the helper data h , the server is able to recover the corresponding Ed-PUF response generated on site, i.e., $\hat{R}$ in Fig. 5.7. This will minimize the false rejection rate due to the bit error probability of PUF response. To authenticate the acquisition source and the data integrity, the server can recalculate the tag using the same HMAC technique as described in Sec. 5.2.4. If the received tags match the recalculated tags,

Event-driven Dynamic Vision Sensor based Physical Unclonable Function for Camera Authentication in Reactive Monitoring System

Fig. 5.8 Flow for activation and de-activation of logic obfuscation upon detection of load capacitance perturbation.

the data is validated to originate from an authorized or a trusted camera and the received event package has not been tampered. Otherwise, the received data is untrustable and a warning signal is raised. Depending on the application requirement, the authentication can be performed selectively based on the detected events or the frequency of authentication can be regulated, e.g., every 10 minutes, to maintain the high event rate of the DVS sensor.

- **Lock Stage:** The *Capacitance Change Detector* module in Fig. 5.5(a) is always on. Once an abrupt or abnormal perturbation of load capacitance is detected, the obfuscation logic is activated to prevent the leakage of actual raw PUF responses. The sensor chip can be unlocked given a correct key, but the server will only respond to key request sent from the authorized processor attached to the sensor and raised by an authorized user (e.g., the user needs to be identified during the key request process). Once a key request is acknowledged by the server, the processor transmits the received e_{obf} and the associated helper data h_{obf} to the server. The response is not transmitted but registered internally as the key under this situation. With the enrolled “ $C'_1 - SADO$ ” mapping, the server can retrieve the random number (as elaborated in the **Authentication Phase**) given e_{obf} and h_{obf} . Thanks to the use of reverse

fuzzy extractor, the retrieved random number is the same as the random number generated by the device on site. Since the server is authorized to know the control signals to all obfuscation gates in the sensor chip, it is able to calculate the correct key from the retrieved random number for the current key request session. The key is sent back to the camera to unlock the sensor chip. The random number for each key request session is triggered by activities in the scene after receiving t_{obf} . Therefore, e_{obf} will be different from session to session. A new key is required to unlock the chip if the obfuscation logic is activated again by another t_{obf} after the chip has been unlocked. It is hard to present or replicate an activity or a series of motion to match an artificially created or previously collected e_{obf} for a key request session. The flow chart in Fig. 5.8 provides an overview of the activation and de-activation of obfuscation logic upon detection of abnormal load capacitance perturbation.

5.3 Performance Evaluation

The proposed system will be evaluated for the following quality of performance. Firstly, as a PUF, the Ed-PUF should have acceptable uniqueness, randomness and reliability. Secondly, the quality of the DVS images and imaging operation should not be appreciably affected by the concurrent PUF operation. Last but not least, the proposed system should have a high level of security with the necessary overhead to solve the challenging problem of coherent detection of sensor data integrity and sensing device authenticity.

5.3.1 Dataset Preparation

Two datasets are used to evaluate the proposed system, which are the baseline PUF database and event data. The baseline PUF CRPs are obtained by using the Cadence design tool to physically layout the DVS-based PUF pixel (see Fig. 5.14, which consists of *Pixel_Core* and M_{13} - M_{15} in Fig. 5.4(b)) in UMC 180nm technology. Post-layout Monte Carlo simulation⁷ of those DVS-based PUF pixels are conducted in a wide range of temperature and supply voltages to obtain the “ C'_1 – *SADO*” database in various environmental settings. The DVS event data is collected by a DAVIS [167] (size: 240×180) in a real scene. DAVIS can perform both DVS and APS imaging at the same time. By disabling the APS mode, address event only data of DVS can be obtained. The camera is used for surveillance purpose as indicated in the proposed use case. In the experiment, a 9 second event stream is generated by having a man walk into the

⁷According to [179], after running Monte Carlo simulation for more than X iterations, the relative error of the probability distribution of possible outcome values can be calculated to be proportional to $X^{-1/2}$. A large X (in my experiments, $X > 10000$) ensures high accuracy (>99%) outcome distribution in the presence of variability in transistor attributes characterized by the foundry model files.

Event-driven Dynamic Vision Sensor based Physical Unclonable Function for Camera Authentication in Reactive Monitoring System

surveillance area, stop to observe, steal a bag and leave. The proposed Ed-PUF is simulated in Matlab by using the post-layout baseline PUF data and the real event data of the DVS camera.

5.3.2 Ed-PUF Performance

Different from existing PUFs, whose final response is solely stimulated by the challenge, the proposed Ed-PUF is triggered by motion in the scene and determined by both a series of activities (that describe a complete or part of an incident) and the challenge. Since the DVS camera has a size of 240×180 pixels, the challenge C is set to 16 bits. The challenge C is randomly generated and is arbitrarily set to be updated after every 1000 responses. The length l of the final PUF response (denoted as R) is set to 511 bits and Q of the modulo counter is set to 32. The final Ed-PUF responses are collected using the experimental setup described in Sec. 5.3.1.

Uniqueness

Uniqueness evaluates the “distinguishability” between the responses of different sensor instances given the same challenge and event data. The uniqueness of the PUF can be estimated by averaging the Hamming distances calculated from a number of responses generated by different pairs of PUF instances with the same challenge-event as follows:

$$Uniqueness = \frac{2}{I(I-1)} \sum_{i=1}^{I-1} \sum_{j=i+1}^I \frac{HD(R_i, R_j)}{l} \quad (5.8)$$

where I and l are the number of sampled instances and bit length of R , respectively.

To evaluate the uniqueness of the proposed Ed-PUF, ten DVS-based PUF instances are simulated by Cadence Spectre[®] in a nominal environment (1.8 V and 27 °C) to obtain the raw *SADO*, which are processed in Matlab using the same set of randomly generated challenges C and the real-captured event data to obtain a set of 511-bit Ed-PUF responses R . 115 final responses are generated from each instance with an event stream of 9s for one external challenge. A total of 5 external challenges are used, resulting in 575 responses per simulated Ed-PUF instance. Therefore, according to Eq. 5.8, the uniqueness can be calculated from the Hamming distances of $\frac{10 \times 9}{2} \times 575 = 25875$ pairs of responses with the same challenge and event stream. Fig. 5.9 shows the histogram of the normalized HD between different pairs of final responses generated by identical event streams and challenges. The mean μ is measured to be 0.5005 while the standard deviation σ is measured to be 0.0223, which are both close to the ideal uniqueness values (ideal $\mu = 0.5$, ideal $\sigma = 0.0221$ for a 511-bit response [2]).

Fig. 5.9 Uniqueness test ($\mu = 0.5005$; $\sigma = 0.0223$). The distribution of normalized HD is an approximate normal distribution.

Reliability

Reliability is evaluated by the BER of responses generated by the same instance given the same challenge and event data in different environmental conditions. Ideally, the reliability should be 100% under the specified range of operating voltage and temperature.

$$Reliability = 1 - BER = 1 - \frac{1}{n} \sum_{i=1}^n \frac{HD(R_0, R_i)}{l} \quad (5.9)$$

where n denotes the number of sampled responses, R_0 and R_i denote the reference response measured at nominal condition and the response measured at the i -th environmental setting, respectively. The reliability of the proposed PUF is tested over the industrial grade of operating temperature, from -45°C to 95°C with a step size of 30°C , as well as $\pm 11\%$ variations (1.6, 1.7, 1.8, 1.9 and 2.0 Volts) in supply voltage using 115 511-bit response pairs. The results are shown in Fig. 5.10, which indicate that the proposed Ed-PUF has > 0.98 and > 0.99 reliability against temperature and supply voltage variations, respectively.

Fig. 5.10 Reliability tests against temperature and supply voltage variations.

Event-driven Dynamic Vision Sensor based Physical Unclonable Function for Camera Authentication in Reactive Monitoring System

Table 5.1 Results of NIST tests.

Statistical	P value	Proportion	Result
Frequency	0.0414	23/23	Pass
BlockFrequency	0.4846	23/23	Pass
CumulativeSums(forward)	0.0151	23/23	Pass
CumulativeSums(backward)	0.2430	22/23	Pass
Runs	0.1866	23/23	Pass
LongestRun	0.0414	22/23	Pass
Rank	0.3115	23/23	Pass
FFT	0.0212	22/23	Pass
NonOverlappingTemplate	0.3925	22/23	Pass
OverlappingTemplate	0.0781	23/23	Pass
ApproximateEntropy	0.2430	22/23	Pass
Serial(forward)	0.4846	23/23	Pass
Serial(backward)	0.7887	22/23	Pass
LinearComplexity	0.0414	23/23	Pass

Randomness

Randomness tests measure the unpredictability of the response. A common method to evaluate randomness is to use the statistical test suite of National Institute of Standards and Technology (NIST). This NIST randomness test contains 15 statistical tests in total. Each test will output its corresponding *p-value* and a proportion rate *PROP*. Together they indicate (in statistical sense) if the input sequence is cryptographically random. The detailed description and requirement of each test and the test suite software can be found in [39]. 470120 response bits were generated from the proposed Ed-PUF and are divided into 23 sub-sequences of 20440 bits each. For the response bitstream to be considered as random, the *p-value* of each statistical test should be larger than 0.01 and the *PROP* should be at least 21/23. The results of 12 statistical tests are shown in Table 5.1 except the Maurer’s “Universal Statistical Test”, “Random Excursions test” and “Random Excursions Variant test”, which will require months of post-layout simulation to collect the required number of response bits to meet the very long sequences of input bits (more than 387840 bits, 1000000 bits and 1000000 bits, respectively) for each sub-sequence in these three tests. The experimental results in Table 5.1 show that the proposed PUF successfully passed all the 12 NIST tests conducted.

Autocorrelation test has also been conducted to indicate the randomness of the proposed Ed-PUF. A good randomness implies no autocorrelation. Autocorrelation test is applied to the above 470120 consecutive generated response bits with lags from 1 to 5000. Fig. 5.11 shows that the autocorrelation factors for all lags are insignificant, and fall into the low ± 0.0029 range

with 95% confidence bound. It shows that the proposed PUF is highly resistant to correlation analysis attack.

Fig. 5.11 Autocorrelation test result.

5.3.3 Collision Test

In the proposed scheme, the PUF operation is triggered by the detected events. The M_{er} transistor of the triggered row will be turned on to reset all the pixels in this row. Consequently, its normal DVS sensing functionality is prohibited for a short time. Although the triggered row is determined by shuffling the row address of the detected event by the output of the LFSR, collision can still happen if there are intensity changes in the scene that happen at the pixels that are about to fire in the triggered row during the brief PUF response generation time. Let P_{DVS} be the probability that a specific row at time t is about to be processed upon firing by an image activity. Two conditions need to be fulfilled in order to cause the address event of fired pixels in this row to be missed as a result of the PUF operation. These conditions are: (1) a PUF operation takes place exactly at this moment and (2) the row that is selected for PUF operation happens to be this row. Let the probabilities of (1) and (2) be P_t and P_{PUF} , respectively. Since these events can be assumed to be independent, the collision probability can be calculated by $P_C = P_t \times P_{PUF} \times P_{DVS}$.

The PUF operation is triggered once in every Q events. Assume the average time to output one event is t_a , and the time to hold the triggered row in reset mode for PUF operation is t_{PUF} . Typically $t_a > 1\mu s$ since the maximum event rate of DVS is around 1 event per μs . Based on the simulation results by Cadence, $t_{PUF} < 1\mu s$. Therefore,

$$P_t = \frac{t_{PUF}}{Q \times t_a} < \frac{1}{Q} \quad (5.10)$$

Event-driven Dynamic Vision Sensor based Physical Unclonable Function for Camera Authentication in Reactive Monitoring System

As mentioned in Sec. 5.2.3, the row that is selected for PUF operation is C'_1 , which is the the rightmost $\lceil \log_2 M \rceil$ bits of C' . Note that the bit length of the LFSR $L > \lceil \log_2 M \rceil + \lceil \log_2 N \rceil$ (see footnote 3). There are 2^L different binary strings in a L -bit word but only $2^{\lceil \log_2 M \rceil}$ distinct combinations can appear in the rightmost $\lceil \log_2 M \rceil$ bits. Each distinct binary pattern of the rightmost $\lceil \log_2 M \rceil$ bits can therefore appear exactly $2^{L-\lceil \log_2 M \rceil}$ times except the all-zero pattern. The occurrence of the all-zero pattern is one less than other patterns because it will not be output by an L -bit maximum length LFSR. If L is large, the probability of occurrence of all-zero pattern and all other patterns in the rightmost $\lceil \log_2 M \rceil$ output bits of the LFSR will be approximately equal, and is given by

$$P_{LFSR} = \frac{2^{L-\lceil \log_2 M \rceil}}{2^L - 1} \quad (5.11)$$

In the experiment, $L = 24$ and $M = 240$, $P_{LFSR} \approx 0.0039$. Since $C' = Address_{event} \oplus LFSR_{out}$, the probability of a specific row being selected by C' in the PUF mode at each triggering moment is $P_{PUF} = P_{LFSR} \approx 0.0039$.

It is important to note that the activities in the scene cannot be predicted. So it is not possible to determine P_{DVS} . However, as $P_{DVS} \leq 1$, the upper bound of the collision probability P_C can be obtained as:

$$\begin{aligned} P_C &= P_t \times P_{PUF} \times P_{DVS} < \frac{1}{Q} \times 0.0039 \times 1 \\ \rightarrow P_C &< 1.22 \times 10^{-4} \quad \text{for } Q = 32 \end{aligned} \quad (5.12)$$

Based on above analysis, we also evaluate the collision by experiment. Assume e_1 is the event that triggers a PUF operation and e_2 is the event detected directly after e_1 in an event package. For e_2 to collide with the PUF operation triggered by e_1 , the following two conditions must be met: (1) the row that performs the PUF operation is the row address of e_2 ; (2) the time difference between e_1 and e_2 should be less than $1 \mu s$. The experimental result shows that there are only 146 collisions found in the testing event stream that has 1881490 events in total. Therefore, the collision probability for this event stream can be approximated as:

$$\frac{146}{1881490} \approx 7.76 \times 10^{-5} < 1.22 \times 10^{-4} \quad (5.13)$$

Collision may lead to a reduced number of detected events compared to normal DVS image sensor without the PUF. Due to the asynchronous property of DVS sensing, the effect introduced by the PUF operation is similar to removing the events in the PUF row while keeping all the other events in the same order from the address events acquired by a normal DVS camera. The

remaining events will form a new event package. If the information conveyed by the new event stream is indiscernibly different from the original event stream, the quality of the DVS image is not degraded. In other words, the contour of the activity detected by the DVS is well preserved.

DVS is frameless but virtual “frames” can be created for visualization purpose by post processing the address events. The “event stream” frames so constructed are “2D histograms of signed (or unsigned, when option *rectifyPolarities* is checked) DVS event counts” [180]. Two methods are typically used to create the “event stream” frames. As shown in Fig. 5.12, the asynchronous DVS events are denoted by blue dots. The “event stream” frames are created by either accumulating all the DVS events over a fixed time interval Δt (Fig. 5.12(a)) or a fixed number of events $\Delta events$ regardless of time (Fig. 5.12(b)) [181]. The intensities of the red and green dots in the “event stream” frames denote the frequencies of “on” and “off” events, respectively accumulated at each specific pixel position.

Fig. 5.12 Construction of “event stream” frames by spatially composing the address-events collected (a) over a fixed period of time; (b) from a fixed number of events;

To visualize the collision, “event stream” frames are created from the original and new event streams by accumulating a fixed number of events. For each position in the DVS sensor array, all its associated events’ polarity (“on”: 1; “off”: -1) will be accumulated to form the pixel value of the “event stream” frame. High value pixel (dominated by “on” events) is colored red while low value pixel (dominated by “off” events) is colored bright green. Central value pixels in the frame are colored black. They are mainly contributed by the static background. Fig. 5.13 shows the frames created from the original event stream and the PUF-in-operation event stream. Collision, if any, can be visually detected by the presence of those red or green dots in the original “event stream” frames without PUF operation but absence in the corresponding “event stream” frames with PUF in operation. Since collision is manifested as missing pixels in the “event stream” frame, denoising filter used to improve the quality of visualization will not remove the collision. The two groups of frames in Fig. 5.13 have invisible difference. Therefore, the PUF operation has negligible or no impact on the visual quality of real-time DVS imaging.

Fig. 5.13 “Event stream” frames (composed by events collected over a fixed interval) made using (a) original event stream without PUF operation (video: https://youtu.be/n_JGVVi2nx8); (b) PUF-in-operation event stream (video: <https://youtu.be/gDvOix4f2c>).

5.3.4 Security and Overhead Analyses

The proposed scheme is designed for reactive monitoring systems. The DVS camera embedded with the Ed-PUF is deployed for area surveillance purposes. As introduced in Sec. 5.2.5, an attacker may try to compromise the system after the camera is deployed. The attacker is assumed to know everything except the “ $C'_1 - SADO$ ” PUF database. In this section, different attack scenarios are analyzed to evaluate the system security⁸.

Man-in-the-middle attack appears frequently in movies and has been demonstrated in [182] to attack traditional surveillance system in a remotely reasonable way. The attacker does not have to access the surveillance camera but can fool the server by wiretapping the transmission. The attacker may insert a customized tap into a live cable so that the data traffic can be routed through the attacker’s device. In this way, the attacker is able to either continue forwarding traffic transparently (to monitor and collect the authentic data) or to generate fake traffic in either direction. However, such an attack is not feasible for the proposed scheme. First of all, the raw PUF response is not transmitted directly but used as the HMAC key to create the authentication tag. As the attacker does not know the embedded PUF response, he/she is not able to calculate a valid tag for a fake event stream which is either modified from authentic data or recorded by a malicious device. Even if the attacker loops the previously collected authentic event data with their associated timestamps modified accordingly to create a seemingly untampered event data, since the tag contains the time information, the tag of a valid event becomes outdated and invalid. Alternatively, the attacker may try to guess the PUF response directly. For a 511-bit PUF response, due to the use of [511, 238, 37] BCH code, a security level equivalent

⁸Automatic security analysis tools like ProVerif [146], CryptoVerif [147], Tamarin [148] or AVISPA [149] can also be used to provide security proof.

to a 238-bit key is achieved, i.e., the probability of correctly guessing the key is 2^{-238} . Since an event stream usually contains multiple event packages, and each package is hashed using different PUF responses due to the different triggered events, the probability to create a valid tag for the whole event stream is actually many orders of magnitude smaller. The PUF response can be set to a longer length to further decrease the success rate of random guessing.

The attacker may try to extract the PUF information from the DVS camera directly. In practice, it is extremely difficult for an attacker to hold a surveillance camera, dismantle it, reassemble it and put it back in the original position without being noticed. Assuming such challenging conditions can all be met by a powerful attacker, due to the tamper-resistant property of PUF, any invasive attempt to probe into the sensor chip internally will damage the subtle disorder property of the PUF response generation mechanism and lead to wrong extracted PUF responses. As shown in Fig. 5.5(b), the raw PUF responses are sent to the off-chip processor directly. It is easier for the attacker to obtain the raw PUF information by controlling the communication interface between the sensor chip and the processor inside the DVS camera. Note that the processor is not programmable after assembly. A possible attack is by wiretapping the connection with the attacker's eavesdropping device to collect the raw PUF response. However, with the logic obfuscation, the sensor chip is locked upon detecting of the load capacitance change caused by such malicious activities. A specific key has to be applied to unlock the chip. Since $r_i \forall i$ in Fig. 5.5 comes from the random Ed-PUF response, which is triggered by unpredictable and irreproducible events, the deobfuscation key is different for each key request session, the attacker, without knowing the PUF " $C'_1 - SADO$ " mapping and the obfuscation logic, cannot calculate the correct key even if he can intercept the event package. If a wrong key is applied, the obfuscated PUF responses measured and collected by the attacker pose no threat to the proposed system.

While it is important to enhance the system security, the introduced system overhead shall also be reasonably minimized to make the scheme practical. The baseline DVS consumes power at couples of mW [166, 167], but the power consumed by the PUF is at μ W level [22, 25]. More importantly, it is only stimulated by motion detected in a fraction of pixels of the entire array at any time. The power overhead is mainly dominated by the always-on *Capacitance Change Detector* module and the extra power dissipated by the HMAC operations. Since this work targets the surveillance camera, there is sufficient power budget from the mains to support these operations. Low-power HMAC, oscillator and counter module are standard primitives in securing data communication. The extra power budget is considered a necessary and fair trade-off to achieve the desirable reliable and secure surveillance data communication. As for the hardware overhead, the area of each pixel cell is measured to be $30 \times 30 \mu\text{m}^2$ with the area mainly dominated by the photodiode and two capacitors. As

Event-driven Dynamic Vision Sensor based Physical Unclonable Function for Camera Authentication in Reactive Monitoring System

Fig. 5.14 Layout of a single pixel using UMC 180nm technology with $C_1 = 240fF$ and $C_2 = 12fF$.

shown in Fig. 5.14, the three additional small transistors M_{13} - M_{15} are easily absorbed into the void of the original DVS pixel area with almost no area overhead. All the other additional circuits for the proposed scheme are shared device-wise. Specifically, the bias V_{ref2} consumes an additional area of $360 \times 22 \mu m^2$ ($= 0.0079 mm^2$) as reported in [169] and the area of a 1×90 Sense Amplifier Array for a 240×180 DVS is measured to be $856.1 \times 12 \mu m^2$ ($= 0.0103 mm^2$); The main digital control logics like *Challenge_Gen*, *Row Decoder*, *Multiplexer* and *Serial to Parallel Converter* are synthesized from Verilog code and placed and routed by the synthesis tool using UMC 180nm standard cell libraries for their area estimation. The total area of this part is reported to be $695 \times 101.7 \mu m^2$ ($= 0.0707 mm^2$). The area of extra circuits to realize the proposed system is around $0.089 mm^2$ ($= 0.0079 + 0.0103 + 0.0707$) while the pixel array of an original 240×180 DVS core is $38.88 (7.2 \times 5.4) mm^2$. Therefore, the area overhead added to a DVS sensor core due to the proposed Ed-PUF is only $0.089/38.88 = 0.23\%$. The area overhead is smaller if those common AER circuitries, i.e., the pixel-wise *Pixel_AER* and the die-wise *Row/Column Handshaking Logic & Row/Column Encoder* and *Row/Column Arbiter Tree*, are also included in the denominator of the area overhead estimation. Lastly, since the PUF operation and the event generation have independent and non-intrusive readout path, no timing overhead is introduced into the 4-phase handshaking protocol. Hence, normal event detection speed will not be influenced by the proposed PUF operation. Thanks to the syndrome construction based reverse fuzzy extractor, the time-consuming BCH decoding has been shifted to the server side. Therefore, the output data rate of DVS, i.e.,

the data rate of the tagged event stream, is limited only by the throughput of HMAC, whose performance bottleneck resides in its underlying hash core [183]. Keccak-512 (one of the SHA-3 finalists) was reported to have a throughput rate higher than 6 Gbps on Virtex 5 or more advanced versions of FPGA implementation [184].

5.3.5 Advantages over APS-based PUF

To the best of my knowledge, the only existing PUF derived from image sensor is [22], which is an APS-based PUF. The work in [22] is mainly about the baseline PUF itself, but the proposed work focuses more on building a comprehensive secure and efficient PUF system. Due to its frame-based characteristic, the raw PUF response of [22] can only be generated by resetting the entire image sensor array, making it categorically unrelated to the sensing environments. Such PUF can only be passively evoked by an input challenge but not actively respond to environmental changes. Actually, most if not all of the existing PUFs in the literature are passive in this sense. In contrast, the authentication of the proposed Ed-PUF is driven mainly by events detected in real time. Such scheme helps to reduce the power dissipation and the amount of redundant data communication. By leveraging on inherent speed and motion detection capability of DVS, tamper-resistant Ed-PUF data tag can be generated for abnormal events without requiring complex post-event processing of image frames, which greatly enhances the system sensitivity and responsiveness. Concurrent PUF response generation and real-time data acquisition of the proposed PUF system cannot be achieved with APS-based PUF without degradation of image quality and throughput due to the inherent constraints of synchronous sensing and global reset in frame-based imaging.

5.4 Summary

This chapter presents a new notion of data-triggered PUF. Based on which, both data and device authentication for reactive monitoring system can be achieved simultaneously. With CCTV displaced by internet-connected digital video recorders in surveillance applications, the loss of “closeness” has expanded the attack horizon and threatened the use of video footage as evidence. If there is no means to ascertain that the footage or its acquisition device have not been tampered, the recorded activities is limited in its utility to law enforcement. The design of a new Ed-PUF system to secure reactive monitoring with the capability to certify if the received video footage is untampered and originated from a trusted camera is presented in this chapter. A new authentication scheme is proposed to make live streaming of surveillance footage secure without being limited by the bandwidth and security of cellular networks. This

Event-driven Dynamic Vision Sensor based Physical Unclonable Function for Camera Authentication in Reactive Monitoring System

is made viable by using the asynchronous address event detected by a DVS camera to trigger the PUF response generation without intervening concurrent acquisition of address events. A HMAC is used to bind each timestamped address event package of an event stream to its PUF response, and syndrome construction based reverse fuzzy extractor technique is utilized to mitigate the PUF reliability problem. The DVS sensor chip is also protected against severing and splicing attacks by a dynamic obfuscation scheme. The advantage of the PUF response being event-driven without impacting event detection rate can be taken to further reduce the bandwidth of data tagging by operational awareness. For example, for DVS cameras deployed for traffic monitoring [165], the tagging can be triggered when the event speed exceeds a pre-defined threshold to provide a tangible evidence of speeding. At present, it is very challenging to fully replace live monitoring by artificial intelligence based video content analysis due to the difficulty to scale the latter to adapt to a broad range of different surveillance scenarios. From this perspective, the proposed Ed-PUF can augment intelligent video content analyzer in proactive video surveillance by providing the credential to attest a low confident alert or a missing alert after the fact.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

IoT is currently experiencing a boost in popularity. 75 billion devices are predicted to be connected to the internet by 2025. With this enormous network of smart “things”, the avenues for hacking a public endpoint are numerous and insidious. A fraction of infective devices can easily blossom into a pandemic in no time. This is what turns a hardware vulnerability into a Gordian knot in the IoT age. Fortunately, the risks have aroused increasing awareness and are better understood by chip makers, designers and security professionals than before. Although addressing them is never easy and requires an ongoing and concerted effort, they are not insurmountable, but rather an arm race determined by the cost-benefit trade-off to pursue a hardware attack or defend. Hence, lightweight security primitive are key to incentivize added defense to any system. PUF, as an emerging hardware security primitive, with its ability to provide a tamper-evident unique/unclonable/unpredictable/reliable “device fingerprint” on demand in a cheap and power-efficient way, has received overwhelming attention from both academia and industry since its inception in 2002. Sensors, as a bedrock that feeds intelligence into IoT nodes, are almost entirely unsecured. Hence they are also the most attack-alluring targets. With vibrant “interaction” among heterogeneous intelligent “things” in the IoT ecosystem, there is a dire need to build a more comprehensive security measure to de-anonymize not only devices, but also users and data so that there is an association of identities among affiliated data, device(s) and/or user(s).

A comprehensive review of PUFs and existing multi-factor authentication methods have been carried out in this thesis. It is found that: 1) PUF offers several unique features that are not directly available from conventional cryptographic methods. The most salient advantages are its tamper awareness and the avoidance of the reliance on non-volatile local storage of a secret key, which is very costly to protect in an IoT node; 2) existing authentication schemes usually

Conclusion and Future Work

perform only single or multiple but separate authentications of user, device or data, with the system security provided by conventional cryptographic methods; 3) although there is high flexibility in extracting PUF responses from original sensor circuit with minimum hardware overhead, no PUF circuit derived directly from sensors has embraced sensing information into the challenge-response mapping. These findings and deficiencies of existing research has motivated us to transform PUF from a mere device identity generator into a unified compound signature of device, user biometrics and sensing data to enable the development of more comprehensive protocol- and system-level protection for end point authentication, digital forensics and reactive monitoring systems.

To this end, three secure and efficient schemes to unify PUF-based device signatures with biometric-based user identity and vision sensor (frame-based and address-event-based) generated data information have been developed. New authentication protocols/schemes are also proposed to support each specific application and completed with security analyses. The main contributions of this thesis can be summarized as follows:

- A PUF-based user-device hash (a.k.a. UDhashing) is proposed and experimentally evaluated. Facial features are extracted as the user signature while the device signature is provided by a strong machine learning resilient PUF. A bio-code that carries both user and device information is generated by fusing the compacted facial features and the PUF response together using the proposed UDhashing scheme. UDhashing comprises four processes, i.e., feature segmentation, random projection, bio-code concatenation and quantization. The main idea is to project the feature segments into a PUF-defined random space. This simple unification scheme effectively eliminates the risk of a locally stored secret seed and template protection challenges faced by the conventional biohashing schemes. A mutual authentication protocol between the end point and the server is also proposed. It allows the end point to authenticate the server before sending any sensitive information, and enables the server to authenticate both the end point device and end point user by the bio-code. The adoption of a strong PUF enables each generated bio-code to be used only once, which eases the protection of bio-codes. Cancelable biometrics can also be achieved since the bio-code leaks no information about the raw biometrics and can be easily reissued if it is reported to be compromised, lost or stolen. The proposed scheme achieves good discriminability, high authentication rate and good tolerance against unreliable PUF responses. Although UDhashing is demonstrated using facial features, it does not exclude the use of other human biometrics for the user registration.
- A PUF-based perceptual data-device hash is developed for digital forensics after the image have gone public. This work is able to authenticate the acquisition device authenticity and the image data integrity, and locate possible tampering of an image, with a target enrolled

device in hand. The proposed hash is time-, data- and device-dependent, and is sensitive to malicious image tampering but otherwise robust against normal image processing like compression, filtering, noising and etc.. A CMOS image sensor based PUF is utilized as the “device key” with one response set forming the Bernoulli random matrix while the other set being the unbiasedRandIntGen() seed. The rotation/scaling-invariant SURF features and adjoint block-based DCT features are extracted as the perceptual image features. The direct data-device hash is obtained by projecting the two kinds of perceptual image features into their corresponding PUF-defined Bernoulli random spaces and concatenating them together. To avoid image feature information leakage through the direct hash due to the use of a weak PUF, a PUF-seeded Knuth Shuffle technique is used to protect the hash. With the target enrolled device in hand, no enrolment phase is needed for the attestation of image integrity. An optimal selection of hash dimension and an adaptive threshold are also proposed to maximize the discrimination between malicious tampering and content-preserving operations. The proposed hash is evaluated to have good perceptual robustness, high tamper detection and location rate in either “tampered only” or “tampered + perceptual processing” cases, and excellent source camera identification performance. The impact of PUF reliability and security analysis are conducted to confirm the viability of the proposed system.

- The first-ever event-driven PUF system for reactive monitoring is proposed, implemented and evaluated. The event-driven PUF is designed from a dynamic vision sensor to avoid redundant and irrelevant background data to be produced, which highly improves the system efficiency. The notion of event-driven PUF is made possible by the unique pixel structure of DVS and its address-event representation mechanism. The differential voltage of each pixel at reset has been analyzed to be highly susceptible to manufacturing process variations. A non-intrusive readout path to access this analog voltage in PUF operation is designed into the original DVS pixel array by adding only three transistors per pixel. The new sensor architecture has the following features: 1) the PUF can only be stimulated by effective (sampled) *Event_Gen* signals and the internal challenge is determined by both the detected event’s addresses and an externally programmable challenge; 2) a hardware lock is designed by logic obfuscation to protect the raw PUF responses. The transmission of the PUF raw response can only be activated by an authorized user for a authorized processor; 3) a PUF-based HMAC scheme is proposed to generate the hash tags, which can be explicitly transmitted over an untrusted channel. An authentication protocol is also specially designed to illustrate the enrolment, authentication and locking/unlocking of the system. In addition to being event-driven, the proposed PUF system is able to validate both device authenticity and event data integrity. The proposed system is evaluated using the raw baseline PUF data and the event data captured by a real DVS camera. The raw baseline PUF data are collected from the post-layout Monte Carlo

simulation of the proposed DVS pixel structure in UMC180nm technology. The experimental results show good PUF performance, negligible collision effect and high security level with reasonably low hardware overhead.

6.2 Future Work

The research works performed and results generated by this thesis provide the basis for some envisaged development and applications of sensor based PUFs. Several improvements can still be made, and new research questions can be raised. In this context, two lines of extension of currently reported results are worthy of further pursued and investigation. They are briefly described here.

6.2.1 Event-driven PUF System Improvements

An event-driven PUF have been proposed and detailed in Chapter 5 of this thesis. Due to the candidature and funding constraint, it is not in time to conduct live tests of a real event-driven PUF system as it requires the costly fabrication of a full-custom integrated circuit of a complete high resolution (of at least 240×180 pixels) DVS pixel array completed with address decoders, row/column arbiters, sense amplifiers, AER logic, bias generator, logic encryption and miscellaneous peripheral supporting modules. The proposed system was currently demonstrated using post-layout simulation data of the event-driven PUF core (*Pixel_Core* + M_{13} - M_{15}) and event data captured by a commercial DVS in the field. While this setup is adequate to prove the practicality and fully extract and evaluate the performances of the proposed scheme, it is difficult to scale for deeper analysis and facilitate massive data collection by the system. Live testing will enable efficient adaptation, testing and experimentation with more complex scenes and different application scenarios. It is certainly time consuming to build such a complete test bed without engineering support, but worth while as a valuable and necessary step towards commercialization. If budget permits, a complete prototype event-driven PUF system-on-chip can be fabricated with the proposed PUF pixel operation readout circuit incorporated into a larger DVS sensor array. The silicon “ $C'_1 - SADO$ ” will be collected as the enrolled PUF database. The fabricated sensor chip will be deployed in the field so that the event-driven PUF performance can be evaluated live using its direct output. Actual timing violations of the PUF operation and DVS event generation can also be analyzed more spontaneously by comparing the “Event stream” frames with and without turning on the PUF operation.

As pointed out in the conclusion of Chapter 5, there is great potential for research into real-time AI-based content analysis based on DVS address events. Only when the probability

that several consecutive address events falling in the classes of interested events exceeds a certain threshold will the PUF operation be triggered. This will extend the usefulness of the proposed event-driven PUF system to proactive video surveillance as well. Due to the high sensitivity of DVS, the detected events could be caused by environmental noises instead of real events. Some detected events may be caused by regular activities in the surveillance area and some malicious activities may be disguised to possess specific characteristics to fool the AI-based content analysis. There are therefore scope for more research into how image content in address-event representation could be denoised, filtered and strengthened based on application and live scene context for training in order to generate more accurate PUF triggers. Due to the analysis of the detected events, processing delay will be inevitably introduced into the system. What is more, the filtering of some “normal” events could be maliciously leveraged by the attackers to hide their evil intention. Therefore, there will be more trade-off optimization to be explored around the system intelligence, latency and security level.

6.2.2 Aging Effect on PUF-based Unification Methods

Another promising research direction is to study and mitigate the potential aging effect on PUF-based unification methods presented in this thesis. Many of the current PUFs (including those used by or proposed in this thesis) evaluate the reliability against only temperature and supply voltage variations. Aging effect has largely been neglected, some for arguable reasons that the cryptographic operations and device authentications are not performed as frequently as normal functionalities. Aging is aggravated by the device scaling phenomena such as negative-bias temperature instability (NBTI), hot-carrier injection (HCI), temperature-dependent dielectric breakdown (TDDB) and electromigration, with NBTI and HCI being the main contributors. Different from the temporal influence introduced by temperature or voltage variations, aging will cause irreversible deterioration of circuit performance over time. While timing guard bands have been conservatively added in each level of design abstraction in existing integrated circuit design flow to ensure that such deterioration will not be perceivable in most general purpose computing devices, some critical paths of new and custom designs may still operate at the verge of the maximum operable clock speed and their flawless operations throughout the intended product lifetime cannot be guaranteed.

There are some works on aging induced PUF performance change in the literature. Since actual aging in the field takes a longer time to produce measurable outcomes, accelerated aging is used to speed up aging effect manifestation. By stressing the device with over-rated temperature, voltage, etc., normal aging process can be sped up. It was found that aging can degrade PUF reliability, but has negligible impact on randomness [185]. Several countermeasures have been proposed to mitigate aging effect, such as selecting ring oscillator

Conclusion and Future Work

(RO) pairs with larger frequency difference, parallelizing aging slope/rate in RO PUF [185, 186], and disabling the PUF circuit when not in use[187, 188]. On the other hand, deliberate aging can also be utilized advantageously to enhance the PUF performance. In [189], aging can either reverse or enlarge the threshold voltage difference of the two cross-coupled inverter PMOS transistors in a SRAM cell to achieve better uniformity or reliability for SRAM PUF. A similar idea was also demonstrated in [190].

Security solutions to IoT applications should take into consideration the lifetime of IoT devices. For most IoT applications, the device life cycle is 10 to 20 years or more, aging effect is definitely an important factor in designing IoT security. As a result, future work in this direction may include:

- Study the “aging tolerance” of specific PUF in an IoT application. A PUF that has smaller performance degradation over a fixed aging period is said to have better “aging tolerance”. The “aging tolerance” of a specific PUF can be simulated using Virtuoso RelXpert Reliability simulator of Cadence Design System®, or further evaluated using accelerated aging techniques based on fabricated ASIC chips or FPGA-based implementations. Environmental and usage factors of a PUF, e.g., temperature, voltage, frequency of use and etc., should be accounted for in the protocol design involving different PUFs to cater for specific IoT applications.
- Propose aging mitigation techniques to deal with the aging problem. After characterizing the aging effect of a PUF, the next step is to find countermeasures to mitigate the effect. For the work presented in Chapter 3, it has good tolerance towards unreliable PUF responses. If the PUF performance degradation rate is kept in an acceptable range after aging over the device lifetime, no or simple reliability enhancement (e.g., majority voting) can be incorporated. For applications that have high requirement on PUF reliability (those in Chapter 4 and Chapter 5), more effective aging countermeasures should be considered in the design of such systems. For example, by design of more aging-resilient PUF, use of circuit-level biasing to reduce the degenerative mechanism, reduction of switching activity to increase recovery time of trapped charge or exploration of more efficient ECC techniques for response reconciliation.

Publications

Journals

- [1] **Y. Zheng**, Y. Cao, and C. H. Chang, “UDhashing: Physical unclonable function based user-device hash for endpoint authentication,” *IEEE Trans. Industrial Electronics (TIE)*, vol. 66, pp. 9559–9570, Jan. 2019.
- [2] **Y. Zheng**, Y. Cao, and C. H. Chang, “A PUF-based data-device hash for tampered image detection and source camera identification,” *IEEE Trans. Inf. Forensics, Security (TIFS)*, vol. 15, pp. 620-634, Jul. 2019.
- [3] **Y. Zheng**, X. Zhao, S. Takashi, Y. Cao, and C. H. Chang, “Ed-PUF: Event-driven physical unclonable function for camera authentication in reactive monitoring system,” *IEEE Trans. Inf. Forensics, Security (TIFS)*, PP. 1-1, Mar. 2020 (Early Access).
- [4] A. Cui, C. H. Chang, W. Zhou, and **Y. Zheng**, “A new PUF based lock and key solution for secure infield testing of cryptographic chips,” *IEEE Trans. Emerging Topics in Computing (TETC)*, Mar. 2019 (Early Access).

Magazine

- [1] 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, pp. 32–62, Aug. 2017.

Conferences

- [1] B. Wang, X. Zhao, **Y. Zheng**, and C. H. Chang, “An in-pixel gain amplifier based event-driven physical unclonable function for CMOS dynamic vision sensors,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Sapporo, Japan, May 2019.
- [2] **Y. Zheng**, Y. Cao, and C. H. Chang, “Facial bihashing based user-device physical unclonable function for bring your own device system (invited paper),” in *Proc. IEEE Int. Conf. Consumer Electronics (ICCE)*, Las Vegas, USA, Jan. 2018 (Invited Paper).
- [3] **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)*, Florence, Italy, May 2018.
- [4] S. S. Dhabu, **Y. Zheng**, W. Liu, and C. H. Chang, “Active IC metering of digital signal processing subsystem with two-tier activation for secure split test,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Florence, Italy, May 2018.
- [5] C. Liu, **Y. Zheng**, and C. H. Chang, “A new write-contention based dual-port SRAM PUF with multiple response bits per cell,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Baltimore, USA, May 2017.
- [6] Y. Cao, C. H. Chang, and **Y. Zheng**, “An energy efficient true random number generator based on current starved ring oscillators,” in *Proc. IEEE Asian Hardware-Oriented Security Trust (AsianHOST)*, Beijing, Oct. 2017 (Cisco Best Paper Award Candidate).
- [7] **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, pp. 1–6, Dec. 2016.

Invited Media Release

- [1] “Can BYOD be as Secure as Company-Owned Devices?” *IEEE Xplore Innovation Spotlight*, 6 Aug. 2018. [online] <https://innovate.ieee.org/innovation-spotlight/biohashing-physical-unclonable-function-byod-authentication-scheme/>.

Bibliography

- [1] 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, pp. 32–62, thirdquarter 2017.
- [2] Y. Zheng, Y. Cao, and C. H. Chang, “UDhashing: Physical unclonable function based user-device hash for endpoint authentication,” *IEEE Trans. Industrial Electronics (TIE)*, vol. 66, pp. 9559–9570, Jan. 2019.
- [3] Y. Zheng, Y. Cao, and C. H. Chang, “A PUF-based data-device hash for tampered image detection and source camera identification,” *IEEE Trans. Inf. Forensics, Secur. (TIFS)*, vol. 15, pp. 620–634, Jul. 2019.
- [4] Y. Zheng, X. Zhao, S. Takashi, Y. Cao, and C. H. Chang, “Ed-PUF: Event-driven physical unclonable function for camera authentication in reactive monitoring system,” *IEEE Trans. Inf. Forensics, Secur. (TIFS)*, pp. 1–1, Mar. 2020.
- [5] R. Maes, P. Tuyls, and I. Verbauwhede, “Low-overhead implementation of a soft decision helper data algorithm for SRAM PUFs,” in *Cryptographic Hardware and Embedded Syst. (CHES 2009)*, pp. 332–347, Springer, 2009.
- [6] M. Guo, “A dynamic vision sensor with direct logarithmic output and full-frame picture-on-demand,” Master’s thesis, Nanyang Technological University, Singapore, 2016.
- [7] K. Ashton, “That ‘Internet of Things’ Thing,” Jun. 2009. [Online]. Available: <https://www.rfidjournal.com/articles/view?4986>.
- [8] Statista, “Internet of Things (IoT) Connected Devices Installed Base Worldwide from 2015 to 2025 (in billions),” Nov. 2016. [Online]. Available: <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/>.
- [9] “Security in the Internet of Things,” *White Paper, Wind River*, Jan. 2015. [Online]. Available: https://www.windriver.com/whitepapers/security-in-the-internet-of-things/wr_security-in-the-internet-of-things.pdf.
- [10] Department for Digital, Culture, Media & Sport, “Code of practice for consumer IoT security,” tech. rep., United Kingdom government., Oct. 2018. [Online]. Available: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/773867/Code_of_Practice_for_Consumer_IoT_Security_October_2018.pdf.

Bibliography

- [11] ETSI, “Etsi releases first globally applicable standard for consumer iot security,” Feb. 2019. [Online]. Available: <https://www.etsi.org/newsroom/press-releases/1549-2019-02-etsi-releases-first-globally-applicable-standard-for-consumer-iot-security>.
- [12] D. A. George, “IoT Manufacturers – What You Need to Know About California’s IoT Law,” Jan. 2020. [Online]. Available: <https://www.natlawreview.com/article/iot-manufacturers-what-you-need-to-know-about-california-s-iot-law>.
- [13] S. Zeitouni, Y. Oren, C. Wachsmann, P. Koeberl, and A.-R. Sadeghi, “Remanence decay side-channel: The PUF case,” *IEEE Trans. Informa. Forensics Security*, vol. 11, pp. 1106–1116, Jun. 2016.
- [14] S. P. Skorobogatov, “Illegitimi Non Carborundum,” Aug. 2011. [Online]. Available: <http://people.csail.mit.edu/rivest/2011-08-15-rivest-CRYPTO-keynote.pdf> (Invited keynote talk given at CRYPTO).
- [15] Cellebrite, “Access Digital Data to Deliver Evidence that Counts,” 2019. [Online]. Available: https://cf-media.cellebrite.com/wp-content/uploads/2017/05/DataSheet_UFEDUltimate_A4_2019_web.pdf.
- [16] B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas, “Silicon physical random functions,” in *Proc. 9th ACM conf. Computer and Commun. security*, (Washington, USA), pp. 148–160, 2002.
- [17] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, “Physical one-way functions,” *Science*, vol. 297, pp. 2026–2030, Sept. 2002.
- [18] S. P. Skorobogatov, “Semi-invasive attacks – A new approach to hardware security analysis,” Tech. Rep. 630, University of Cambridge, Apr. 2005.
- [19] G. E. Suh and S. Devadas, “Physical unclonable functions for device authentication and secret key generation,” in *Proc. 44th ACM/IEEE Design Autom. Conf. (DAC)*, (San Diego, USA), pp. 9–14, Jun. 2007.
- [20] D. Lim *et al.*, “Extracting secret keys from integrated circuits,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, pp. 1200–1205, Dec. 2005.
- [21] D. E. Holcomb, W. P. Burleson, and K. Fu, “Power-up SRAM state as an identifying fingerprint and source of true random numbers,” *IEEE Trans. Comput.*, vol. 58, pp. 1198–1210, Nov 2009.
- [22] 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, pp. 2629–2640, Nov. 2015.
- [23] 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. 2016 IEEE Asian Hardware-Oriented Secur. Trust (AsianHOST) Symposium*, (Taiwan), pp. 1–6, Dec. 2016.
- [24] Y. Cao, C. 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*, pp. 1–10, Aug. 2018.
 - [25] C. Liu, Y. Cao, and C. H. Chang, “ACRO-PUF: A low-power, reliable and aging-resilient current starved inverter-based ring oscillator physical unclonable function,” *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 64, pp. 3138–3149, Dec. 2017.
 - [26] C. Liu, Y. Cao, and C. H. Chang, “Low-power, lightweight and reliability-enhanced current starved inverter based RO PUFs,” in *Proc. 2016 IEEE Asia Pacific Conf. Circuits Syst. (APCCAS)*, (Jeju, South Korea), pp. 646–649, IEEE, Oct. 2016.
 - [27] E. Peterson, “Developing Tamper-Resistant Designs with Zynq UltraScale+ Devices,” Aug. 2018. [Online]. Available: https://www.xilinx.com/support/documentation/application_notes/xapp1323-zynq-usp-tamper-resistant-designs.pdf.
 - [28] Intrinsic ID, “NXP adds PUF Anti-Cloning Technology to Its Next Generation SmartMX2 Microcontroller,” Sept. 2016. [Online]. Available: <https://www.intrinsic-id.com/nxp-adds-puf-anti-cloning-technology-next-generation-smartmx2-microcontroller/>.
 - [29] X. Guo, D. M. Jacobson, Y. Yang, A. J. Drew, and B. M. Rosenberg, “Applying circuit delay-based physically unclonable functions (PUFs) for masking operation of memory-based PUFs to resist invasive and clone attacks,” Oct. 2017. US Patent 9,787,480.
 - [30] W. Suberg, “Accenture, Thales to Debut Blockchain Tool for Aerospace Supply Chains,” Jul. 2018. [Online]. Available: <https://cointelegraph.com/news/accenture-thales-to-debut-blockchain-tool-for-aerospace-supply-chains>.
 - [31] Y. Shoukry, P. Martin, P. Tabuada, and M. Srivastava, “Non-invasive spoofing attacks for anti-lock braking systems,” in *Proc. Int. Workshop Cryptographic Hardware Embedded Syst. (CHES)*, (Santa Barbara, CA, USA), pp. 55–72, Aug. 2013.
 - [32] ISO/IEC Joint Technical Committee 1, “ISO/IEC DIS 20897 Security requirements and test methods for physically unclonable functions for generating non-stored security parameters,” Oct. 2019. [Online]. Available: <https://www.iso.org/obp/ui#iso:std:iso-iec:20897:-1:dis:ed-1:v1:en>.
 - [33] A. Maiti, J. Casarona, L. McHale, and P. Schaumont, “A large scale characterization of RO-PUF,” in *Proc. IEEE Int. Symp. Hardware-Oriented Secur. Trust (HOST)*, (Anaheim, USA), pp. 94–99, Jun. 2010.
 - [34] 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)*, (Rio, Brazil), pp. 2325–2328, May 2011.

Bibliography

- [35] R. van den Berg, “Entropy analysis of physical unclonable functions,” Master’s thesis, Eindhoven University of Technol., Eindhoven, NL, Aug. 2012.
- [36] R. Van Den Berg, B. Skoric, and V. van der Leest, “Bias-based modeling and entropy analysis of PUFs,” in *Proc. 3rd Int. Workshop Trustworthy Embedded Devices (TrustED)*, (Berlin, Germany), pp. 13–20, Nov. 2013.
- [37] M.-D. Yu, R. Sowell, A. Singh, D. M’Raïhi, and S. Devadas, “Performance metrics and empirical results of a PUF cryptographic key generation ASIC,” in *Proc. IEEE Int. Symp. Hardware-Oriented Secur. Trust (HOST)*, (San Francisco, USA), pp. 108–115, Jun. 2012.
- [38] G. Marsaglia, “The marsaglia random number CD-ROM with the diehard battery of tests of randomness,” 1995.
- [39] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, *et al.*, “A statistical test suite for random and pseudorandom number generators for cryptographic applications,” Tech. Rep. 800-22 Rev 1a, Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Apr. 2010. [Online]. Available: <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-22r1a.pdf>.
- [40] J. Zaman and R. Ghosh, “A review study of NIST statistical test suite: Development of an indigenous computer package,” *arXiv preprint arXiv:1208.5740*, Aug. 2012.
- [41] J. C. Moreira and P. G. Farrell, *Essentials of Error-Control Coding*. John Wiley & Sons, 2006.
- [42] R. W. Hamming, “Error detecting and error correcting codes,” *The Bell System Tech. J.*, vol. 29, pp. 147–160, Apr. 1950.
- [43] T. K. Moon, *Error Correction Coding: Mathematical Methods and Algorithms*. John Wiley & Sons, 2005.
- [44] C. Böhm, M. Hofer, and W. Pribyl, “A microcontroller SRAM-PUF,” in *Proc. 2011 5th Int. Conf. Network System Security*, (Milan, Italy), pp. 269–273, IEEE, Sept. 2011.
- [45] 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 Trustable Comput.*, pp. 39–81, Springer, 2016.
- [46] I. S. Reed and X. Chen, “BCH codes,” in *Error-Control Coding for Data Networks*, pp. 189–231, Springer, 1999.
- [47] U. Rührmair and D. E. Holcomb, “PUFs at a glance,” in *Proc. Design, Autom. & Test in Europe Conf. & Exhibition (DATE)*, (Dresden, Germany), pp. 1–6, Mar. 2014.
- [48] U. Rührmair *et al.*, “Modeling attacks on physical unclonable functions,” in *Proc. 17th ACM conf. Comput. Commun. Security*, (Chicago, Illinois, USA), pp. 237–249, Oct. 2010.

-
- [49] Y. Gao, D. C. Ranasinghe, S. F. Al-Sarawi, O. Kavehei, and D. Abbott, "Emerging physical unclonable functions with nanotechnology," *IEEE Access*, vol. 4, pp. 61–80, Feb. 2016.
- [50] S. S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, "The butterfly PUF protecting IP on every FPGA," in *Proc. IEEE Int. Workshop Hardware-Oriented Secur. Trust (HOST)*, (Anaheim, USA), pp. 67–70, Jun. 2008.
- [51] P. Simons, E. van der Sluis, and V. van der Leest, "Buskeeper PUFs, a promising alternative to D flip-flop PUFs," in *Proc. IEEE Int. Symp. Hardware-Oriented Secur. Trust (HOST)*, (San Francisco, USA), pp. 7–12, Jun. 2012.
- [52] Y. Su, J. Holleman, and B. P. Otis, "A digital 1.6 pJ/bit chip identification circuit using process variations," *IEEE J. of Solid-State Circuits*, vol. 43, pp. 69–77, Jan. 2008.
- [53] Y. Wang *et al.*, "Flash memory for ubiquitous hardware security functions: True random number generation and device fingerprints," in *Proc. IEEE Symp. Secur. and Privacy*, (San Francisco, USA), pp. 33–47, May 2012.
- [54] L. Zhang, Z. H. Kong, and C. H. Chang, "PCKGen: a phase change memory based cryptographic key generator," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, (Beijing, China), pp. 1444–1447, May. 2013.
- [55] L. Zhang, Z. H. Kong, C. H. Chang, A. Cabrini, and G. Torelli, "Exploiting process variations and programming sensitivity of phase change memory for reconfigurable physical unclonable functions," *IEEE Trans. Inform. Forensics Security*, vol. 9, pp. 921–932, Apr. 2014.
- [56] V. Van der Leest, G.-J. Schrijen, H. Handschuh, and P. Tuyls, "Hardware intrinsic security from D flip-flops," in *Proc. 5th ACM Workshop Scalable Trusted Comput.*, (Chicago, Illinois, USA), pp. 53–62, Oct. 2010.
- [57] G. E. Suh, C. W. O'Donnell, and S. Devadas, "Aegis: A single-chip secure processor," *IEEE Des. & Test Comput.*, vol. 24, pp. 570–580, Dec. 2007.
- [58] K. Lofstrom, W. R. Daasch, and D. Taylor, "IC identification circuit using device mismatch," in *Proc. IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers (ISSCC)*, (San Francisco, USA), pp. 372–373, Aug. 2000.
- [59] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," in *Proc. Int. Workshop Cryptographic Hardware Embedded Syst. (CHES)*, (Vienna, Austria), pp. 63–80, Sept. 2007.
- [60] C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," *Proc. IEEE*, vol. 102, pp. 1126–1141, May 2014.

Bibliography

- [61] R. Bernardini and R. Rinaldo, “Helper-less physically unclonable functions and chip authentication,” in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, (Florence, Italy), pp. 8193–8197, May 2014.
- [62] Y. Sheffer, G. Zorn, H. Tschofenig, and S. Fluhrer, “An EAP authentication method based on the encrypted key exchange (EKE) protocol,” tech. rep., RFC 6124, Feb. 2011. [Online]. Available: <http://www.rfc-editor.org/info/rfc6124>.
- [63] B. Schneier, *Applied cryptography: protocols, algorithms, and source code in C*, 2nd ed. John Wiley and Sons, NY, Nov. 1995.
- [64] A. Van Herrewege and others, “Reverse fuzzy extractors: Enabling lightweight mutual authentication for PUF-enabled RFIDs,” in *Proc. Int. Conf. Financial Cryptography and Data Security*, (Kralendijk, Bonaire), pp. 374–389, Feb. 2012.
- [65] X. Boyen, “Reusable cryptographic fuzzy extractors,” in *Proc. the 11th ACM Conf. Comput. Commun. Secur. (CCS)*, (Washington, USA), pp. 82–91, ACM, Oct. 2004.
- [66] J. Delvaux, *Security analysis of PUF-based key generation and entity authentication*. PhD thesis, School of Cyber Science and Engineering, Shanghai Jiaotong University, Jun. 2017. [Online]. Available: <https://www.esat.kuleuven.be/cosic/publications/thesis-290.pdf>.
- [67] A. Schaller, T. Stanko, B. Škorić, and S. Katzenbeisser, “Eliminating leakage in reverse fuzzy extractors,” *IEEE Trans. Inf. Forensics Secur.*, vol. 13, pp. 954–964, Apr. 2018.
- [68] M.-D. M. Yu, D. M’Raihi, R. Sowell, and S. Devadas, “Lightweight and secure PUF key storage using limits of machine learning,” in *Proc. Int. Workshop Cryptographic Hardware Embedded Syst. (CHES)*, (Nara, Japan), pp. 358–373, Sept. 2011.
- [69] D. Suzuki and K. Shimizu, “The glitch PUF: A new delay-PUF architecture exploiting glitch shapes,” in *Proc. Int. Workshop Cryptographic Hardware Embedded Syst. (CHES)*, (Santa Barbara, USA), pp. 366–382, Aug. 2010.
- [70] U. Rührmair, “Oblivious transfer based on physical unclonable functions,” in *Proc. Int. Conf. Trust Trustworthy Comput.*, (Berlin, Germany), pp. 430–440, Jun. 2010.
- [71] D. Lim, “Extracting secret keys from integrated circuits,” Master’s thesis, Massachusetts Inst. of Technol, Cambridge, MA, USA, May. 2004.
- [72] G. Hospodar, R. Maes, and I. Verbauwhede, “Machine learning attacks on 65nm Arbiter PUFs: Accurate modeling poses strict bounds on usability,” in *Proc. Int. Workshop Inform. Forensics Secur. (WIFS)*, (Tenerife, Spain), pp. 37–42, Dec. 2012.
- [73] U. Rührmair *et al.*, “PUF modeling attacks on simulated and silicon data,” *IEEE Trans. Inform. Forensics Security*, vol. 8, pp. 1876–1891, Aug. 2013.

-
- [74] M. Majzoobi, M. Rostami, F. Koushanfar, D. S. Wallach, and S. Devadas, "Slender PUF protocol: A lightweight, robust, and secure authentication by substring matching," in *Proc. IEEE Symp. Secur. Privacy Workshop (SPW)*, (San Francisco, USA), pp. 33–44, May 2012.
- [75] M. Rostami, M. Majzoobi, F. Koushanfar, D. S. Wallach, and S. Devadas, "Robust and reverse-engineering resilient PUF authentication and key-exchange by substring matching," *IEEE Trans. Emerging Topics in Comput. (TETC)*, vol. 2, pp. 37–49, Jan. 2014.
- [76] G. T. Becker, "On the pitfalls of using arbiter-PUFs as building blocks," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, pp. 1295–1307, Apr. 2015.
- [77] M. Nabeel, S. Kerr, X. Ding, and E. Bertino, "Authentication and key management for advanced metering infrastructures utilizing physically unclonable functions," in *Proc. IEEE 3rd Int. Conf. Smart Grid Commun. (SmartGridComm)*, (Tainan, Taiwan), pp. 324–329, Nov. 2012.
- [78] V. Seferian, R. Kanj, A. Chehab, and A. Kayssi, "PUF and ID-based key distribution security framework for advanced metering infrastructures," in *Proc. 2014 IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, (Venice, Italy), pp. 933–938, Nov. 2014.
- [79] U. S. Premarathne, I. Khalil, and M. Atiquzzaman, "Secure and reliable surveillance over cognitive radio sensor networks in smart grid," *Pervasive Mobile Comput.*, vol. 22, pp. 3–15, May 2015.
- [80] M. Potkonjak, S. Meguerdichian, and J. L. Wong, "Trusted sensors and remote sensing," in *Proc. Sensors, IEEE (ICSENS)*, (Kona, USA), pp. 1104–1107, Nov. 2010.
- [81] K. Rosenfeld, E. Gavas, and R. Karri, "Sensor physical unclonable functions," in *Proc. IEEE Int. Symp. Hardware-Oriented Secur. Trust (HOST)*, (Anaheim, USA), pp. 112–117, Jun. 2010.
- [82] S. Meguerdichian and M. Potkonjak, "Security primitives and protocols for ultra low power sensor systems," in *Proc. Sensors, IEEE (ICSENS)*, (Limerick, Ireland), pp. 1225–1227, Oct. 2011.
- [83] J. Rajendran, J. Tang, and R. Karri, "Securing pressure measurements using sensor PUFs," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, (Montreal, Canada), pp. 1330–1333, May 2016.
- [84] M. Spencer *et al.*, "Demonstration of integrated micro-electro-mechanical relay circuits for VLSI applications," *IEEE J. Solid-State Circuits*, vol. 46, pp. 308–320, Jan. 2011.
- [85] F. Chen *et al.*, "Integrated circuit design with NEM relays," in *Proc. Int. Conf. Comput.-Aided Design*, (San Jose, CA, USA), pp. 750–757, Nov. 2008.

Bibliography

- [86] R. Nathanael *et al.*, “Four-terminal-relay body-biasing schemes for complementary logic circuits,” *IEEE Electron Device Lett.*, vol. 31, pp. 890–892, Aug. 2010.
- [87] VMware, “End-User Device Registration,” Jun. 2019. [Online]. Available: <https://docs.vmware.com/en/VMware-Workspace-ONE-UEM/9.7/vmware-airwatch-guides-97/GUID-AW97-EndUserDeviceRegistration.html>.
- [88] 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>.
- [89] SolarWinds, “User Device Tracker.” [Online]. Available: <https://www.solarwinds.com/user-device-tracker>.
- [90] S. D. Upp, “Device and/or user authentication for network access,” May 8 2008. US Patent App. 11/556,408.
- [91] J. Kaippallimalil and Y. Xia, “User and device authentication in broadband networks,” Aug. 18 2015. US Patent 9,112,909.
- [92] R. Hayton, “User and device authentication in enterprise systems,” Apr. 18 2017. US Patent 9,628,448.
- [93] 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 (TrustED)*, (Denver, USA), pp. 3–13, Oct. 2015.
- [94] Y. Guo and A. Tyagi, “Voice-based user-device physical unclonable functions for mobile device authentication,” *J. Hardware Syst. Security*, vol. 1, pp. 18–37, May 2017.
- [95] K. et al., “Secure and reliable biometric access control for resource-constrained systems and IoT,” *arXiv preprint arXiv:1803.09710*, 2018.
- [96] R. Davarzani, S. Mozaffari, and K. Yaghmaie, “Perceptual image hashing using center-symmetric local binary patterns,” *Multimedia Tools Appl.*, vol. 75, pp. 4639–4667, Mar. 2016.
- [97] 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, pp. 3387–3402, Jul. 2018.
- [98] 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, pp. 1047–1055, May 2018.
- [99] 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, pp. 123–133, Jan. 2017.

-
- [100] L. Wen, H. Qi, and S. Lyu, "Contrast enhancement estimation for digital image forensics," *ACM Trans. Multimedia Comput., Commun., Appl. (TOMM)*, vol. 14, p. 49, May 2018.
- [101] P. Korus and J. Huang, "Improved tampering localization in digital image forensics based on maximal entropy random walk," *IEEE Signal Process. Lett.*, vol. 23, pp. 169–173, Dec. 2016.
- [102] 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.
- [103] X. Lv and Z. J. Wang, "Perceptual image hashing based on shape contexts and local feature points," *IEEE Trans. Inf. Forensics Security*, vol. 7, pp. 1081–1093, Jun. 2012.
- [104] X. Wang *et al.*, "A visual model-based perceptual image hash for content authentication," *IEEE Trans. Inf. Forensics Security*, vol. 10, pp. 1336–1349, Jul. 2015.
- [105] 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, pp. 200–214, Jan 2016.
- [106] P. Meerwald, C. Koidl, and A. Uhl, "Attack on "watermarking method based on significant difference of wavelet coefficient quantization"," *IEEE Trans. Multimedia*, vol. 11, pp. 1037–1041, Aug. 2009.
- [107] T. Qazi *et al.*, "Survey on blind image forgery detection," *IET Image Process.*, vol. 7, pp. 660–670, Oct. 2013.
- [108] S. E. Quadir *et al.*, "A survey on chip to system reverse engineering," *ACM J. Emerging Technol. Comput. Syst. (JETC)*, vol. 13, p. 6, Dec. 2016.
- [109] 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), pp. 99–103, Apr. 2015.
- [110] 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/>.
- [111] 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.
- [112] 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), pp. 2660–2666, Mar. 2017.

Bibliography

- [113] 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), pp. 1848–1854, Jul. 2017.
- [114] J. Lukáš, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Trans. Inf. Forensics Security*, vol. 1, pp. 205–214, Jun. 2006.
- [115] M. Chen, J. Fridrich, M. Goljan, and J. Lukáš, "Determining image origin and integrity using sensor noise," *IEEE Trans. Inf. Forensics Security*, vol. 3, pp. 74–90, Mar. 2008.
- [116] D. Valsesia, G. Coluccia, T. Bianchi, and E. Magli, "Large-scale image retrieval based on compressed camera identification," *IEEE Trans. Multimedia*, vol. 17, pp. 1439–1449, Sept. 2015.
- [117] D. Valsesia, G. Coluccia, T. Bianchi, *et al.*, "User authentication via PRNU-based physical unclonable functions," *IEEE Trans. Inf. Forensics Security*, vol. 12, pp. 1941–1956, Aug. 2017.
- [118] 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 Security*, vol. 12, pp. 392–404, Feb. 2017.
- [119] "Rolling updates on coronavirus disease (COVID-19)," Mar. 2020. Accessed on: Mar. 20, 2020. [Online]. Available: <https://www.who.int/emergencies/diseases/novel-coronavirus-2019/events-as-they-happen>.
- [120] A. Milan, "What countries are on lockdown to combat coronavirus?," Mar. 2020. Accessed on: Mar. 20, 2020. [Online]. Available: <https://metro.co.uk/2020/03/19/countries-lockdown-combat-covid-19-12423352/>.
- [121] S. Dasgupta and A. Gupta, "An elementary proof of the Johnson-Lindenstrauss lemma," *Int. Comput. Science Institute, Tech. Report*, vol. 22, no. 1, pp. 1–5, 1999.
- [122] S. S. Vempala, *The Random Projection Method*, vol. DIMACS-65. American Mathematical Soc., 2005.
- [123] A. B. Teoh, Y. W. Kuan, and S. Lee, "Cancellable biometrics and annotations on biohash," *Pattern Recognition*, vol. 41, pp. 2034–2044, Jun. 2008.
- [124] 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, pp. 1877–1893, Sept. 2011.
- [125] 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, pp. 2795–2805, Apr. 2018.

- [126] Y. Liu and D. Hatzinakos, "Biohashing for human acoustic signature based on random projection," *Canadian J. Electr. Computer Eng.*, vol. 38, pp. 266–273, Summer 2015.
- [127] G.-Y. Chen, M. Gan, F. Ding, and C. P. Chen, "Modified Gram–Schmidt method-based variable projection algorithm for separable nonlinear models," *IEEE Trans. Neural Networks Learning Syst.*, vol. 30, pp. 2410–2418, Aug. 2018.
- [128] 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, pp. 87–100, Jul. 2005.
- [129] Å. Björck and C. C. Paige, "Loss and recapture of orthogonality in the modified Gram–Schmidt algorithm," *SIAM J. Matrix anal. Appl.*, vol. 13, no. 1, pp. 176–190, 1992.
- [130] A. Kong, K.-H. Cheung, D. Zhang, M. Kamel, and J. You, "An analysis of biohashing and its variants," *Pattern Recognition*, vol. 39, pp. 1359–1368, Dec. 2005.
- [131] Y. Zheng, Y. Cao, and C. H. Chang, "Facial biohashing based user-device physical unclonable function for bring your own device security," in *Proc. IEEE Int. Conf. Consum. Electron.*, (Las Vegas, USA), pp. 1–6, Mar. 2018.
- [132] V. Štruc and N. Pavešić, "The complete gabor-fisher classifier for robust face recognition," *EURASIP J. Advances in Signal Process.*, vol. 2010, p. 31, May 2010.
- [133] 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)*, (Florence), May 2018.
- [134] B. Mamandipoor, D. Ramasamy, and U. Madhow, "Newtonized orthogonal matching pursuit: Frequency estimation over the continuum," *IEEE Trans. Signal Process.*, vol. 64, pp. 5066–5081, Oct. 2016.
- [135] M. Cui and S. Prasad, "Sparse representation-based classification: Orthogonal least squares or orthogonal matching pursuit?," *Pattern Recognition Lett.*, vol. 84, pp. 120–126, Aug. 2016.
- [136] 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, pp. 210–227, Feb 2009.
- [137] M. Yu *et al.*, "A lockdown technique to prevent machine learning on PUFs for lightweight authentication," *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 2, pp. 146–159, July 2016.
- [138] 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)*, (KaoHsiung, Taiwan), pp. 33–36, IEEE, Nov. 2014.

Bibliography

- [139] A. Bogdanov *et al.*, “SPONGENT: A lightweight hash function,” in *Proc. Int. Workshop Cryptographic Hardware. Embedded Syst.*, pp. 312–325, Springer, 2011.
- [140] 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 Secur. (TIFS)*, pp. 1–1, Sept. 2018.
- [141] F. S. Samaria and A. C. Harter, “Parameterisation of a stochastic model for human face identification,” in *Proc. 2nd IEEE Workshop. Appl. Comput. Vision*, (Sarasota, USA), pp. 138–142, Dec. 1994.
- [142] Yale, “The Extended Yale Face Database B,” 2001. [Online]. Available: <http://vision.ucsd.edu/~iskwak/ExtYaleDatabase/ExtYaleB.html>.
- [143] K. Nandakumar and A. K. Jain, “Biometric template protection: Bridging the performance gap between theory and practice,” *IEEE Signal Process. Mag.*, vol. 32, pp. 88–100, Aug. 2015.
- [144] W. Kim, S. Suh, and J.-J. Han, “Face liveness detection from a single image via diffusion speed model,” *IEEE Trans. Image Processing*, vol. 24, pp. 2456–2465, Aug. 2015.
- [145] J. Määttä, A. Hadid, and M. Pietikäinen, “Face spoofing detection from single images using texture and local shape analysis,” *IET biometrics*, vol. 1, pp. 3–10, Mar. 2012.
- [146] B. Blanchet, B. Smyth, V. Cheval, and M. Sylvestre, “ProVerif 2.00: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial,” tech. rep., May 2018. Accessed on: Mar. 16, 2020. [Online]. Available: <https://prosecco.gforge.inria.fr/personal/bblanche/proverif/manual.pdf>.
- [147] B. Blanchet, “CryptoVerif: A Computationally-Sound Security Protocol Verifier,” tech. rep., Inria Paris, France, Nov. 2017. Accessed on: Mar. 16, 2020. [Online]. Available: <https://prosecco.gforge.inria.fr/personal/bblanche/cryptoverif/cryptoverif.pdf>.
- [148] “Tamarin-Prover Manual: Security Protocol Analysis in the Symbolic Model,” tech. rep., The Tamarin Team, Jun. 2019. Accessed on: Mar. 16, 2020. [Online]. Available: https://tamarin-prover.github.io/manual/book/001_introduction.html.
- [149] “AVISPA v1.1 User Manual: Automated Validation of Internet Security Protocols and Applications,” tech. rep., The AVISPA Team, Jun. 2006. Accessed on: Mar. 16, 2020. [Online]. Available: <http://www.avispa-project.org/>.
- [150] 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.
- [151] 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 Secur. Trust (AsianHOST) Symposium*, (Yilan, Taiwan), pp. 1–6, Dec. 2016.

-
- [152] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. vision image understanding*, vol. 110, pp. 346–359, Jun. 2008.
- [153] 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), pp. I–I, Apr. 2001.
- [154] E. Oyallon and J. Rabin, "An analysis of the SURF method," *Image Proc. Line*, vol. 5, pp. 176–218, May 2015.
- [155] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. comput. vision*, vol. 60, pp. 91–110, Nov. 2004.
- [156] M. Zuliani, "RANSAC for dummies," *Vision Research Lab, University of California, Santa Barbara*, Jul. 2014.
- [157] P. Scholar, "A survey: digital image watermarking techniques," *Int. J. Signal Process. Image Process. Pattern Recognit*, vol. 7, no. 6, pp. 111–124, 2014.
- [158] 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), pp. 1–5, May 2018.
- [159] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, "The Johnson-Lindenstrauss lemma meets compressed sensing," *preprint*, vol. 100, pp. 1–9, Jan. 2006.
- [160] D. Achlioptas, "Database-friendly random projections," in *Proc. 20th ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems*, pp. 274–281, May 2001.
- [161] S. Eberz, K. Rasmussen, V. Lenders, and I. Martinovic, "Preventing lunchtime attacks: Fighting insider threats with eye movement biometrics," Feb. 2015.
- [162] 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), pp. 422–426, Jul. 2013.
- [163] 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, pp. 55–63, Jan. 2013.
- [164] D. Sunehra and A. Bano, "An intelligent surveillance with cloud storage for home security," in *Proc. IEEE Ann. India Conf. (INDICON)*, (Pune, India), pp. 1–6, Dec. 2014.
- [165] T. Semertzidis, K. Dimitropoulos, A. Koutsia, and N. Grammalidis, "Video sensor network for real-time traffic monitoring and surveillance," *IET Intell. Transp. Syst.*, vol. 4, pp. 103–112, Jun. 2010.

Bibliography

- [166] P. Lichtsteiner, C. Posch, and T. Delbruck, “A 128×128 120 dB 15 μ s latency asynchronous temporal contrast vision sensor,” *IEEE J. Solid-State Circuits*, vol. 43, pp. 566–576, Feb. 2008.
- [167] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, “A 240×180 130 dB 3 μ s latency global shutter spatiotemporal vision sensor,” *IEEE J. Solid-State Circuits*, vol. 49, pp. 2333–2341, Oct. 2014.
- [168] aiCTX, “DyapCNN — the world’s first 1M neuron, event-driven neuromorphic AI processor for vision processing,” Apr. 2019. [Online]. Available: https://aictx.ai/news/dynapcnn_april2019/.
- [169] M. Yang, S.-C. Liu, C. Li, and T. Delbruck, “Addressable current reference array with 170dB dynamic range,” in *Proc. 2012 IEEE Int. Symp. Circuits Syst. (ISCAS)*, (South Korea), pp. 3110–3113, IEEE, May 2012.
- [170] K. Sanborn, D. Ma, and V. Ivanov, “A sub-1-v low-noise bandgap voltage reference,” *IEEE J. Solid-State Circuits (JSSC)*, vol. 42, pp. 2466–2481, Nov. 2007.
- [171] T. Stöhr, M. Alt, A. Hetzel, and J. Koehl, “Analysis, reduction and avoidance of crosstalk on VLSI chips,” in *Proc. 1998 Int. Symp. Physical Design*, (Monterey, USA), pp. 211–218, Apr. 1998.
- [172] M. Yasin, J. J. Rajendran, O. Sinanoglu, and R. Karri, “On improving the security of logic locking,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, pp. 1411–1424, Dec. 2015.
- [173] K. Shamsi, M. Li, T. Meade, Z. Zhao, D. Z. Pan, and Y. Jin, “Cyclic obfuscation for creating SAT-unresolvable circuits,” in *Proc. Great Lakes Symp. VLSI 2017*, (NY, USA), pp. 173–178, ACM, May 2017.
- [174] Y. Tokunaga, S. Sakiyama, A. Matsumoto, and S. Dosho, “An on-chip CMOS relaxation oscillator with voltage averaging feedback,” *IEEE J. Solid-State Circuits*, vol. 45, pp. 1150–1158, Jun. 2010.
- [175] Y. Liu, S. Chen, M. Nakayama, and K. Watanabe, “Limitations of a relaxation oscillator in capacitance measurements,” *IEEE Trans. Instrumentation Measurement*, vol. 49, pp. 980–983, Oct. 2000.
- [176] M. Bellare, “New proofs for NMAC and HMAC: Security without collision-resistance,” in *Ann. Int. Cryptology Conf.*, (Santa Barbara, USA), pp. 602–619, Springer, Aug. 2006.
- [177] Y. Zheng, Y. Cao, and C. Chang, “A new event-driven dynamic vision sensor based physical unclonable function for camera authentication in reactive monitoring system,” in *Proc. 2016 IEEE Asian Hardware-Oriented Secur. Trust (AsianHOST) Symposium*, (Taiwan), pp. 1–6, Dec. 2016.

-
- [178] B. Wang, X. Zhao, Y. Zheng, and C. H. Chang, "An in-pixel gain amplifier based event-driven physical unclonable function for CMOS dynamic vision sensors," in *2019 IEEE Int. Symp. Circuits Syst. (ISCAS)*, (Sapporo, Japan), pp. 1–5, May 2019.
- [179] "UMC 180NM Mixed-Mode and RFCMOS 1.8V Twin-Well MOSFET Monte Carlo Mismatch SPICE Model Document," Tech. Rep. Version 1.2 Phase 4, UMC, Uppsala, Sweden, Feb. 2009.
- [180] T. Delbruck, "User Guide-jAER." [Online]. Available: https://docs.google.com/document/d/1fb7VA8tdoxuYqZfrPfT46_wiT1isQZwTHgX8O22dJ0Q/edit.
- [181] S. Bamford, "The matlab version of aedatools," Nov. 2019. [Online]. Available: <https://github.com/simbamford/AedatTools/>.
- [182] Z. Banks and E. Van Albert, "Looping Surveillance Cameras Like in the Movies," May. 2015. [Online]. Available: <https://www.youtube.com/watch?v=RoOqznZUCII>.
- [183] H. E. Michail, G. S. Athanasiou, V. Kelefouras, G. Theodoridis, and C. E. Goutis, "On the exploitation of a high-throughput SHA-256 FPGA design for HMAC," *ACM Trans. Reconfigurable Technol. Syst. (TRETs)*, vol. 5, no. 1, p. 2, 2012.
- [184] K. Latif, M. M. Rao, A. Aziz, and A. Mahboob, "Efficient hardware implementations and hardware performance evaluation of SHA-3 finalists," in *Proc. Third SHA-3 Candidate Conf.*, (Washington, DC), Mar. 2012.
- [185] A. Maiti and P. Schaumont, "The impact of aging on a physical unclonable function," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, pp. 1854–1864, Sept. 2014.
- [186] M. T. Rahman, D. Forte, F. Rahman, and M. Tehranipoor, "A pair selection algorithm for robust RO-PUF against environmental variations and aging," in *Proc. 33rd IEEE Int. Conf. Comput. Design (ICCD)*, (NY, USA), pp. 415–418, Oct. 2015.
- [187] S. R. Sahoo, K. Sudeendra Kumar, K. Mahapatra, and A. Swain, "A novel aging tolerant RO-PUF for low power application," Dec. 2016.
- [188] M. T. Rahman, F. Rahman, D. Forte, and M. Tehranipoor, "An aging-resistant RO-PUF for reliable key generation," *IEEE Trans. Emerging Topics in Comput.*, vol. 4, pp. 335–348, Sept. 2016.
- [189] A. Garg and T. T. Kim, "Design of SRAM PUF with improved uniformity and reliability utilizing device aging effect," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, (Melbourne, Australia), pp. 1941–1944, Jun. 2014.
- [190] T. Xu and M. Potkonjak, "Stable and secure delay-based physical unclonable functions using device aging," in *Proc. 2015 IEEE Int. Symp. Circuits Syst. (ISCAS)*, (Lisbon, Portugal), pp. 33–36, IEEE, May 2015.