Computer Fraud and Security
ISSN (online): 1873-7056

An Efficient Post-Processing Method for SSL-PUF in MEC Security Authentication

Cheng Chen[1*], Qian Zhang[1,2], and Zhibin Feng[1]

[1]Department of Basic Courses, Rocket Force University of Engineering, Xi'an 710025, China
[2] Experimental Training Base, National University of Defense Technology, Xi'an 710106, China
Corresponding Author's Email: cchen316@163.com

Abstract: Computation offloading is a key technology in mobile edge computing (MEC) that addresses the performance and energy constraints faced by mobile devices when handling computationally intensive tasks. Identity authentication for computation offloading is a critical issue as it ensures data security and user identity legitimate verification. Semiconductor superlattice physical unclonable functions (SSL-PUFs) are unique physical characteristics based on semiconductor superlattice materials, which can be used for secure authentication and encrypted communication in edge computing with wide applications in security authentication. However, the adoption of SSL-PUF in computation offloading for MEC applications faces two practical challenges: insufficient alignment accuracy of response signals and poor stability of SSL-PUF response signals. To address these two issues, an efficient post-processing algorithm specifically designed for SSL-PUF has been proposed. This algorithm consists of two steps. The first step involves aligning the response signals of SSL-PUF using a sequence alignment algorithm based on preset sequence, which significantly reduces the intra-chip Hamming distance of SSL-PUF. Then, a data fusion algorithm combining time majority voting mechanism is used to filter out erroneous response data, thereby improving the accuracy of SSL-PUF response signals. Experimental results demonstrate that after applying the proposed post-processing algorithm, the randomness of the signals remains largely unaffected. The maximum bit error rate of SSL-PUF response signals is reduced by 34.33%, and the average intra-chip Hamming distance decreases from 12% to 4.9%. The reliability of SSL-PUF is significantly enhanced, making it promising for secure identity authentication in mobile edge computing.

Keywords: SSL-PUF; Secure authentication; mobile edge computing; post-processing.

INTRODUCTION

The advent of the "Internet of Everything" era has significantly advanced the prosperity of the Internet of Things (IoT) due to the low latency and high bandwidth offered by 5G networks [1,2]. However, this growth has also introduced more complex IoT security issues [3]. Compute offloading plays a crucial role in IoT's prosperity by enhancing performance, reducing latency, optimizing network resource utilization, enabling complex data processing, fostering service innovation, and strengthening security [4–6]. As a key technology in Mobile Edge Computing (MEC), it addresses the performance and energy constraints faced by mobile devices in executing computation-intensive tasks [7]. During the process of edge computing, there is a significant amount of network data transmission between IoT devices and clouds. It is crucial to pay special attention to the privacy and security of user data to prevent data leakage and malicious exploitation. Therefore, identity authentication in edge computing is a vital aspect that ensures the legitimacy of user identities and the privacy of data [8]. However, IoT edge devices are often resource-constrained devices deployed extensively. Traditional encryption algorithms may have certain limitations when used in such devices. For example, the non-volatile memory (NVM) used to store keys in the device may be vulnerable to side-channel attacks, leading to the risk of key replication [9,10]. Additionally, asymmetric encryption and decryption often require high computational costs, making it challenging to implement in resource-constrained MEC [11].

Physical Unclonable Functions (PUFs) leverage inherent randomness introduced during manufacturing to provide a unique "fingerprint" or trust anchor for physical entities [12,13]. PUFs act as "hardware fingerprints," generated through challenge-response mechanisms (CRPs) that are bound to the PUF [14]. Due to uncontrolla-ble manufacturing variations, even the designer cannot replicate an identical PUF, making it truly unclonable in a physical sense. By extracting keys on demand from reliable and random physical systems rather than storing them in non-volatile memory, the security of IoT edge computing devices is significantly enhanced with the incorporation of PUFs. In recent years, scholars at home and abroad have conducted indepth research on PUFs

and their applications in the field of security. They have proposed various types of PUFs and corresponding applications of PUFs in the context of IoT and mobile edge computing [15,16]. In 2012, Zhang et al. [17] introduced GaAs/$Al_{0.45}Ga_{0.55}As$ semiconductor superlattice (SSL), marking the first time that superlattice chaotic current oscillations were achieved at room temperature. This breakthrough allowed superlattices to move beyond cryogenic laboratories, paving the way for their practical development. Chen et al. [18] suggested incorporating deterministic physical functional properties from semiconductor superlattice devices into the PUF cryptographic theory domain, aiming to enhance PUF cryptographic capabilities using the unique properties of superlattice materials. Later that year, Wu et al. [19] reported the SSL-PUF, for secure communication.

The SSL-PUF is a novel type of PUF that utilizes the inherent physical properties of semiconductor superlattice materials to enhance secure authentication and key generation processes [20]. This technology demonstrates robustness against various attack vectors, including brute force attacks, birthday attacks, and cloning attacks [21]. However, the SSL-PUF faces two practical challenges in realworld applications: insufficient alignment accuracy of the response signal and poor stability of the SSL-PUF response signal. This paper proposes a post-processing algorithm based on precise sequence alignment algorithm and time majority voting to address the response characteristics of SSL-PUF. This algorithm effectively enhances the alignment and accuracy of the SSL-PUF response signal, significantly improving its stability. The maximum bit error rate is reduced by 34.33%, and the average intra-chip Hamming distance drops from 12% to 4.9%. This practical improvement in SSL-PUF technology optimizes response signal processing, thereby increasing its reliability and practical value.

SSL-PUF INFORMATION

Semiconductor superlattices are grown using molecular beam epitaxy (MBE), a semiconductor material growth technique that allows the precise growth of nanoscale semiconductor crystal layers with atomiclevel accuracy. The superlattice consists of 50 periods of weakly coupled potential wells (GaAs) and barriers ($Al_{0.45}Ga_{0.55}As$), sandwiched between two silicon-based GaAs layers to form an n+-n-n+ diode structure [22], as shown in Figure 1(a). Due to the phenomenon of cascaded resonant tunneling, the superlattice forms a multi-degree-of-freedom nonlinear system [23–25], capable of exhibiting a one-to-one correspondence between input challenges and output responses under specific bias voltages. Even slight changes in the input signal can result in completely different output responses.

Semiconductor superlattice devices are typical analog devices that are driven by analog input signals. Unlike conventional PUFs, superlattices require the use of analog-to-digital converters (ADCs) and digital-to-analog converters (DACs) to facilitate the challenge-response process, as shown in Figure 1(b). Specifically, the digital driving sequence (challenge signal) is converted into an analog signal to drive the superlattice operation using DAC. The analog output signal (response signal) from the superlattice is then sampled and digitized by an ADC to obtain a random sequence for further processing. Therefore, compared to regular PUFs, the intra-chip Hamming distance of an SSL-PUF using ADC sampling is much higher.

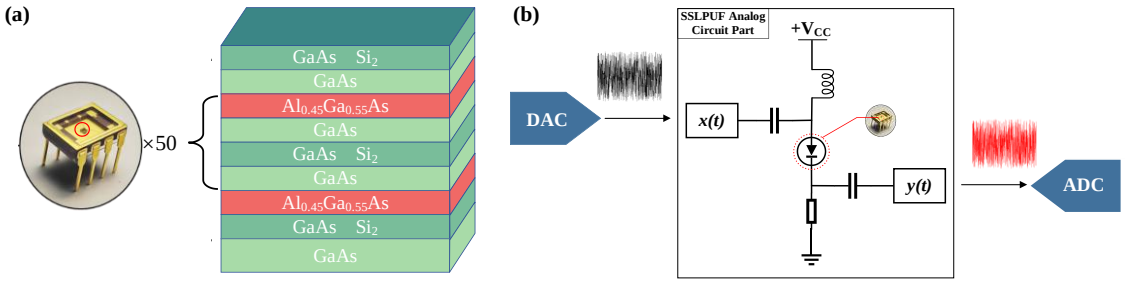

Figure 1. (a) Schematic of the SSL-PUF architecture; (b) Application circuit of the SSL-PUF.

METHODS

PRECISE SEQUENCE ALIGNMENT ALGORITHM

Frame synchronization algorithms are widely used in the mobile communications industry [26]. Effective frame synchronization algorithms enable low-latency and high-precision synchronization, thereby enhancing

communication reliability. To address the issue of response misalignment in SSL-PUF, the concept of frame synchroni-zation is introduced. Firstly, we define the concept of a check sequence. Due to the unique challenge-response mechanism of PUFs, a fixed frame header cannot be directly set. Instead, a specific challenge signal is used, which, when input to the PUF, produces a specific response signal. This specific response is defined as the check sequence, as shown in Equation (1).

$$\{s_1, s_2, ..., s_i\} = PUF(\{c_1, c_2, ..., c_i\}) \tag{1}$$

where $\{s_1, s_2, ..., s_i\}$ denotes the check sequence, and $\{c_1, c_2, ..., c_i\}$ represents the predefined challenge, which is a preset random number. The check sequence is typically determined during the first registration of the SSL-PUF. Upon initial use of the SSL-PUF, a random sequence is generated as the preset challenge signal for verification. This random sequence is combined with a DC bias signal to form the input to the SSL-PUF. The raw response obtained under the influence of the SSL-PUF is considered as a candidate verification pattern. The candidate pattern from the first sam-pling is designated as the standard verification pattern, which is subsequently used for calculating the intra-die Hamming distance in later verifications. As shown in Figure 2, the superlattice input signal consists of a DC bias signal, a predefined challenge signal (frame header), and a challenge signal.

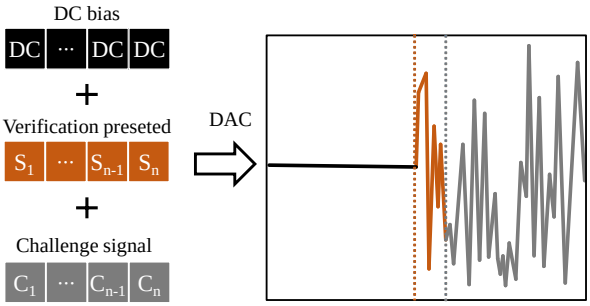

Figure 2. Schematic diagram of the SSL-PUF challenge-response structure.

Under the input signal, a corresponding response signal is generated, which also comprises a DC bias signal, a check sequence, and a response signal. Check reliability is used to describe the ratio of differing bits between the check sequence and the unknown sequence segment to the total number of bits in the check sequence, facili-tating the positioning of the check sequence, as shown in Equation (2).

$$\vartheta(s, l) = \frac{\sum_i^n s_i \square \ l_i}{len(s_i)} \tag{2}$$

Here, l_i represents the unknown sequence, s_i represents the check sequence, and $len(s_i)$ denotes the length of s. By traversing the original response, the position of the check sequence can be located, thereby aligning the response sequence. As shown in Figure 3, this is a schematic diagram of the overall process of the sequence alignment algorithm. First, an input signal is generated in the order of the DC bias signal, the predefined check signal, and the response signal. This input signal is then converted into a suitable challenge signal via a DAC module and fed into the SSL-PUF. The original response signal from the SSL-PUF is converted into a binary sequence using an ADC module. According to prior analysis, this binary response sequence should also comprise three parts: the DC bias interval, the check sequence, and the response signal. A prestored check sequence template is used to traverse the binary sequence, while simultaneously calculating the check reliability. This yields a check reliability sequence. The position corresponding to the maximum value in the check reliability sequence marks the alignment position of the original response. Subsequently, removing the check sequence and the DC bias interval yields the aligned response sequence.

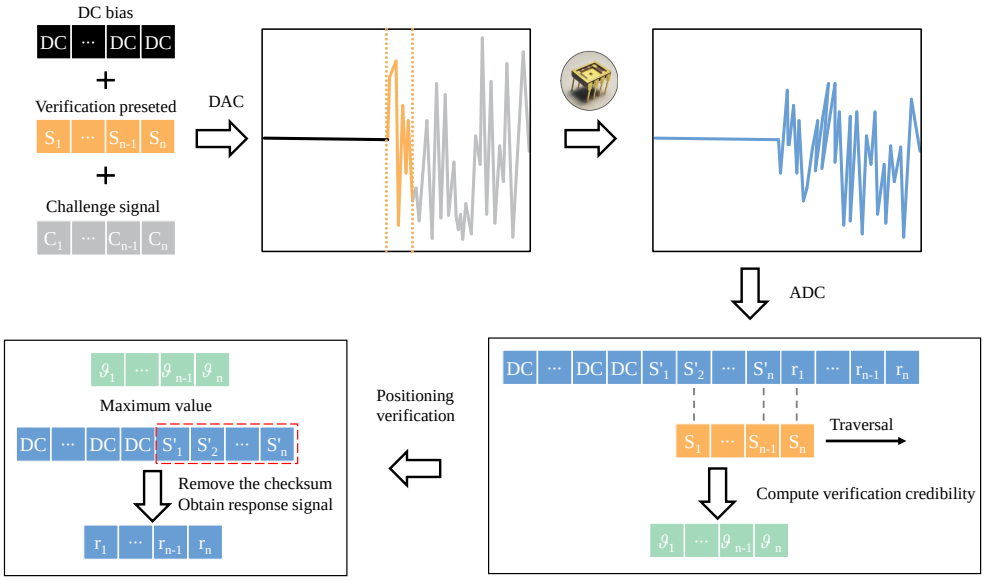

Figure 3. Flowchart of the Precise Alignment Algorithm Based on Frame Synchronization.

TIME MAJORITY VOTING ALGORITHM

Similar to other PUFs, precisely aligning SSL-PUF response signals alone cannot reduce the error rate to an acceptable level, as ADC collection may introduce random noise [27]. A time majority voting (TMV) post-processing method effectively suppresses external instability and random noise in PUF responses. To optimize hardware implementation, registers and accumulators are used to design the algorithm, ensuring high execution efficiency of the TMV algorithm.

Figure 4 illustrates the schematic diagram of the algorithm used in this study, which is based on time majority voting. The algorithm consists of two steps. Firstly, n sets of response signals are repetitively collected, where each set has m bits of effective digits and is aligned bit by bit. Next, each bit of the m-bit data corresponds to an n-bit accumulator, and the n sets of response signals are sequentially processed through the accumulators. When the accumulated value of each bit exceeds n/2, the output is 1, otherwise it is 0. After n rounds of collection, the final output data is obtained as the fused response signal output of this stage.

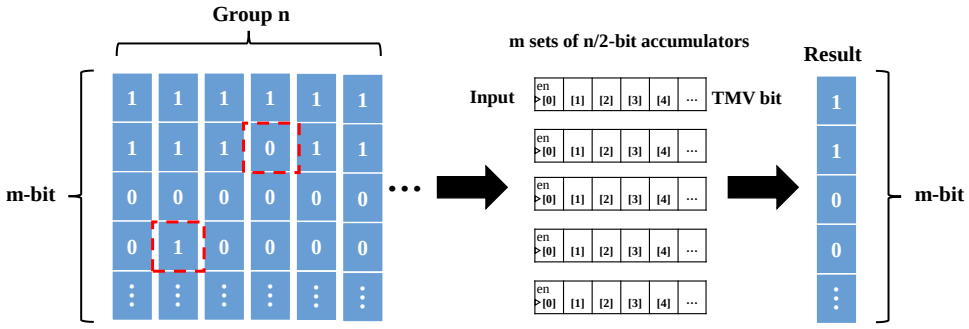

Figure 4. The flowchart for the TMV algorithm implementation.

EXPERIMENT AND ANALYSIS

The Keithley precision measurement DC power supply provides stable DC bias voltage for the SSL-PUF. The input signal for the SSL-PUF is generated by the Tabor P25812B arbitrary waveform generator, and the original response signal is captured by the Teledyne HDO9104-MS oscilloscope. Instruments are connected to a computer via Ethernet, and signal generation and capture are controlled using SCPI commands in MATLAB. The post-processing algorithm verification is also conducted in MATLAB.

EFFECT OF POST-PROCESSING ON SSL-PUF RANDOMNESS

True random number tests commonly use information entropy to assess randomness. Entropy measures disorder and randomness, with min-entropy being a conservative estimation method calculating the lower bound of entropy. The NIST 800-90B standard is used for min-entropy estimation [28]. Given that SSL-PUF responses are not independent and identically distributed, both post-processed and pre-processed responses were tested with 1,000,000 samples each. Results in Table 1 show a slight decrease in min-entropy after processing, but overall, it remains similar, indicating that the post-processing algorithm has minimal impact on the randomness of SSL-PUF.

Table 1. Minimum entropy test of raw response and entropy extracted response.

Methods	RAW	After processing
MCV	0.92393643092000644	0.85745646844658138
Collision	0.82943456119897307	0.78656846546464546
Markov	0.90689232722887891	0.91486464834846348
Compression	0.60961820346704154	0.58464648453189431
t-tuple	0.61051191939745042	0.59464184634643486
LRS	0.96502873355136076	0.92484348434864135
MultiMCW	0.71719177125745914	0.70154641464867484
Lag	0.77453372179186875	0.78464684346841318
MultiMMC	0.71719184382032497	0.67423151897434496
LZ78Y	0.71719182598530484	0.72487434134148413
Minimum	0.60961820346704154	0.58464648453189431

COMPARISON OF POST-PROCESSING ALGORITHM EFFECTS

We take the initial response signal as the reference signal for calculating the intra-chip Hamming distance (HDintra) in the entire experiment, which is a fundamental concept in information theory [29,30] to measure the similarity between two responses. Then, we repeatedly stimulated the same SSL-PUF with the same challenge signal 200 times and collected the corresponding response signals to ensure the accuracy of the experiment. The SSL-PUF response data without any post-processing is referred to as raw data. We applied a high-precision sequence alignment algorithm based on check subsequence for the first processing step on the raw data. Then, we used the Temporal Majority Voting (TMV) data fusion algorithm for the second processing step on the data that had undergone the first processing. We compared the error rates of the raw SSL-PUF response data with the data that had undergone the two processing steps.

The statistics of the intra-chip Hamming distance for the 200 original challenge response signals are shown in Figure 5(a). The raw SSL-PUF response even had a maximum intra-chip Hamming distance of 40%, and the distribution of most data points was relatively scattered, with most falling between 5% and 23%. The average intra-chip Hamming distance at this stage was 12%. This misalignment issue caused the observed results. Figure 5(b) shows the statistics of the intra-chip Hamming dis-tance after applying the high-precision sequence alignment algorithm based on check subsequences. The Hamming distance was concentrated between 4% and 18%, with an average intra-chip Hamming distance of 8.2%. This significantly improved the stability of the SSL-PUF. Finally, we validated the TMV data fusion algorithm. After the data fusion, the response data distribution became more concentrated, ranging from 4% to 10%, with an average intra-chip Hamming distance of 4.9%, as showed in Figure5(c). In Figure 5(d), it can be observed that the maximum HDintra decreased to 16.84% and 10.64% after

applying the high-precision sequence alignment algorithm and TMV data fusion processing, respectively, compared to the original response.

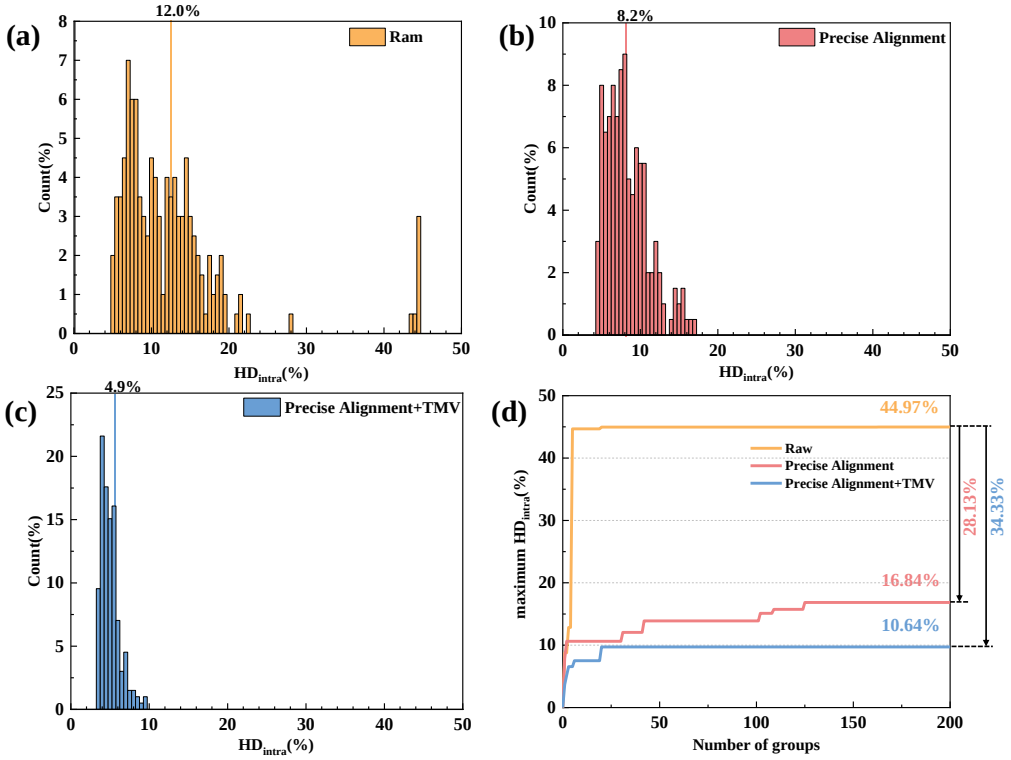

Figure 5. (a) HDintra statistics of the original sequences; (b) HDintra statistics after processing with the precise alignment algorithm; (c) HDintra statistics after processing with the precise alignment algorithm and TMV data fusion; (d) Comparison of the maximum HDintra after no processing and after the two post-processing algorithms.

DISCUSSION

From an application perspective, error correction algorithms are often needed as post-processing to correct erroneous data in response signals. The cost of error correction algorithms is typically correlated with the maximum HDintra of the PUF. A larger HDintra implies a higher computational cost for error correction. After applying the sequence alignment algorithm, the maximum HDintra of the response decreases from 44.97% to 16.84% compared to the original response. Therefore, the proposed sequence alignment algorithm effectively addresses the issue of sequence misalignment in the original response. Additionally, by comparing the maximum HDintra values, incorporating the TMV data fusion algorithm on top of the sequence alignment algorithm further reduces the HDintra to 10.64%. Thus, the data fusion algorithm contributes to improving the stability of the SSL-PUF response. However, this result is still relatively high compared to other PUFs. In future work, we will explore alternative approaches to further reduce the intra-chip Hamming distance of SSL-PUFs and enhance response stability.

CONCLUSIONS

Identity security authentication for computation offloading is a crucial aspect of mobile edge computing. The SSL-PUF can provide higher security in edge computing for identity authentication due to its nonlinear physical properties. However, it suffers from issues related to inadequate alignment accuracy and stability of response signals. To address this problem, this paper proposes an efficient post-processing algorithm to improve the stability of SSL-PUFs. Experimental results demonstrate that the maximum intra-chip Hamming distance, with the help of the preset sequence for precise alignment, decreases from 44.97% to 16.84%, effectively resolving response misalignment. Furthermore, after applying the TMV data fusion algorithm, the maximum intra-chip Hamming distance further decreases to 10.64%, and the average intra-chip Hamming distance reduces to 4.9%,

significantly enhancing the stability of SSL-PUF devices. Although the intra-chip Hamming distance of SSL-PUF has been reduced to a level where it can be eliminated by fuzzy extractors, there is still a noticeable gap in stability compared to other PUFs. Therefore, further efforts will be devoted to optimizing the stability of SSL-PUFs and promoting their applications.

DECLARATION OF CONFLICTING INTERESTS

The author(s) declared no potential conflicts of interest with respect to the research, author-ship, and/or publication of this article.

DATA SHARING AGREEMENT

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

FUNDING

The author(s) received no financial support for the research, authorship, and/or publication of this article.

REFRENCES

[1] W. Shi, W. Xu, X. You, C. Zhao, and K. Wei, “Intelligent Reflection Enabling Technologies for Integrated and Green In-ternet-of-Everything Beyond 5G: Communication, Sensing, and Security,” IEEE Wirel. Commun., vol. 30, no. 2, pp. 147–154, Apr. 2023, doi: 10.1109/MWC.018.2100717.

[2] W. Wei, R. Yang, H. Gu, W. Zhao, C. Chen, and S. Wan, “Multi-objective optimization for resource allocation in vehicular cloud computing networks,” IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 12, pp. 25536–25545, 2021, doi: 10.1109/tits.2021.3091321.

[3] W. Iqbal, H. Abbas, M. Daneshmand, B. Rauf, and Y. A. Bangash, “An In-Depth Analysis of IoT Security Requirements, Challenges, and Their Countermeasures via Software-Defined Security,” IEEE Internet Things J., vol. 7, no. 10, pp. 10250–10276, Oct. 2020, doi: 10.1109/JIOT.2020.2997651.

[4] H. Lin, S. Zeadally, Z. Chen, H. Labiod, and L. Wang, “A survey on computation offloading modeling for edge computing,” J. Netw. Comput. Appl., vol. 169, p. 102781, Nov. 2020, doi: 10.1016/j.jnca.2020.102781.

[5] T. Zheng, J. Wan, J. Zhang, C. Jiang, and G. Jia, “A Survey of Computation Offloading in Edge Computing,” in 2020 Inter-national Conference on Computer, Information and Telecommunication Systems (CITS), Hangzhou, China: IEEE, Oct. 2020, pp. 1–6. doi: 10.1109/CITS49457.2020.9232457.

[6] C. Feng, P. Han, X. Zhang, B. Yang, Y. Liu, and L. Guo, “Computation offloading in mobile edge computing networks: A survey,” J. Netw. Comput. Appl., vol. 202, p. 103366, Jun. 2022, doi: 10.1016/j.jnca.2022.103366.

[7] L. Dong, M. N. Satpute, J. Shan, B. Liu, Y. Yu, and T. Yan, “Computation Offloading for Mobile-Edge Computing with Multi-user,” in 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Jul. 2019, pp. 841–850. doi: 10.1109/ICDCS.2019.00088.

[8] T. Li, X. He, S. Jiang, and J. Liu, “A survey of privacy-preserving offloading methods in mobile-edge computing,” J. Netw. Comput. Appl., vol. 203, p. 103395, Jul. 2022, doi: 10.1016/j.jnca.2022.103395.

[9] Y. Gao, S. F. Al-Sarawi, and D. Abbott, “Physical unclonable functions,” Nat Electron, vol. 3, no. 2, Art. no. 2, Feb. 2020, doi: 10.1038/s41928-020-0372-5.

[10] A. Al-Meer and S. Al-Kuwari, “Physical Unclonable Functions (PUF) for IoT Devices,” Acm Comput. Surv., vol. 55, no. 14s, pp. 1–31, Dec. 2023, doi: 10.1145/3591464.

[11] S. M. A. Huda and S. Moh, “Survey on computation offloading in UAV-Enabled mobile edge computing,” J. Netw. Comput. Appl., vol. 201, p. 103341, May 2022, doi: 10.1016/j.jnca.2022.103341.

[12] D. Zhong et al., “Twin physically unclonable functions based on aligned carbon nanotube arrays,” Nat. Electron., 2022, doi: 10.1038/s41928-022-00787-x.

[13] Y. Liu et al., "A Novel Physical Unclonable Function Based on Silver Nanowire Networks," Adv. Funct. Mater., vol. n/a, no. n/a, p. 2304758, Aug. 2023, doi: 10.1002/adfm.202304758.

[14] Y. Cao et al., "Entropy Sources Based on Silicon Chips: True Random Number Generator and Physical Unclonable Func-tion," Entropy, vol. 24, no. 11, Art. no. 11, Oct. 2022, doi: 10.3390/e24111566.

[15] D. Kwon and Y. Park, "Design of Secure and Efficient Authentication Protocol for Edge Computing-Based Augmented Reality Environments," Electronics, vol. 13, no. 3, Art. no. 3, Jan. 2024, doi: 10.3390/electronics13030551.

[16] H. Yıldız, M. Cenk, and E. Onur, "PLGAKD: A PUF-Based Lightweight Group Authentication and Key Distribution Pro-tocol," IEEE Internet of Things Journal, vol. 8, no. 7, pp. 5682–5696, Apr. 2021, doi: 10.1109/JIOT.2020.3032757.

[17] Y. Huang, W. Li, W. Ma, H. Qin, and Y. Zhang, "Experimental observation of spontaneous chaotic current oscillations in GaAs/Al0.45Ga0.55As superlattices at room temperature," Chinese Sci. Bull., vol. 57, no. 17, pp. 2070–2072, Jun. 2012, doi: 10.1007/s11434-012-5198-8.

[18] X. Tong, X. Chen, and S. Xu, "Advances in superlattice cryptography research," Chinese Sci. Bull., vol. 65, no. 2–3, pp. 108–116, Jan. 2020, doi: 10.1360/TB-2019-0291.

[19] H. Wu et al., "An experimental demonstration of long-haul public-channel key distribution using matched superlattice physical unclonable function pairs," Science Bulletin, vol. 65, no. 11, pp. 879–882, Jun. 2020, doi: 10.1016/j.scib.2020.02.029.

[20] L. Xu et al., "An SSL-PUF Based Access Authentication and Key Distribution Scheme for the Space–Air–Ground Integrated Network," Entropy, vol. 25, no. 5, p. 760, May 2023, doi: 10.3390/e25050760.

[21] X. Li, J. Xie, L. Xu, H. Wu, R. Shi, and H. Feng, "Key space estimation and security analysis of superlattice physical un-clonable function," Microelectronics Journal, vol. 151, p. 106320, Sep. 2024, doi: 10.1016/j.mejo.2024.106320.

[22] W. Li et al., "Fast Physical Random-Number Generation Based on Room-Temperature Chaotic Oscillations in Weakly Coupled Superlattices," Phys. Rev. Lett., vol. 111, no. 4, p. 044102, Jul. 2013, doi: 10.1103/PhysRevLett.111.044102.

[23] Y. Huang, W. Li, W. Ma, H. Qin, H. T. Grahn, and Y. Zhang, "Spontaneous quasi-periodic current self-oscillations in a weakly coupled GaAs/(Al,Ga)As superlattice at room temperature," Appl. Phys. Lett., vol. 102, no. 24, p. 242107, Jun. 2013, doi: 10.1063/1.4811358.

[24] Y. Huang et al., "Experimental evidence for coherence resonance in a noise-driven GaAs/AlAs superlattice," Europhys. Lett., vol. 105, no. 4, p. 47005, Feb. 2014, doi: 10.1209/0295-5075/105/47005.

[25] W. Li et al., "Chaos synchronization in networks of semiconductor superlattices," Europhys. Lett., vol. 112, no. 3, p. 30007, Nov. 2015, doi: 10.1209/0295-5075/112/30007.

[26] D. Nikolaidis, "Novel Minimalist Hardware Architecture for Long Sync Word Frame Synchronization and Payload Cap-ture," Electronics, vol. 13, no. 17, p. 3372, Aug. 2024, doi: 10.3390/electronics13173372.

[27] Y. Shifman, A. Miller, O. Keren, Y. Weizman, and J. Shor, "An SRAM-Based PUF With a Capacitive Digital Preselection for a 1E-9 Key Error Probability," IEEE Trans. Circuits Syst. I, vol. 67, no. 12, pp. 4855–4868, Dec. 2020, doi: 10.1109/TCSI.2020.2996772.

[28] J. Liu et al., "Min-entropy estimation for semiconductor superlattice true random number generators," Sci Rep, vol. 12, no. 1, p. 2948, Feb. 2022, doi: 10.1038/s41598-022-06815-2.

[29] J. Shallit, "Hamming distance for conjugates," Discrete Math., vol. 309, no. 12, pp. 4197–4199, Jun. 2009, doi: 10.1016/j.disc.2008.11.001.

[30] B. Manthey and R. Reischuk, "The intractability of computing the Hamming distance," Theoret. Comput. Sci., vol. 337, no. 1–3, pp. 331–346, Jun. 2005, doi: 10.1016/j.tcs.2005.02.002.