

Analog-Inspired Hardware Security: A Low-Energy Solution for IoT Trusted Communications

Samuel Ellicott, Michael Kines, Waleed Khalil

The Ohio State University

Columbus, Ohio

ellicott.4@osu.edu , kines.1@osu.edu, khalil.18@osu.edu

Yu Qi, Abdullah Kurtoglu, Hossein Miri Lavasani

Case Western Reserve University

Cleveland, Ohio

yxq248@case.edu, axk1214@case.edu, sxm1243@case.edu

Abstract—With the proliferation of connected internet of things (IoT) devices, trusted communications between such devices is an increasing concern. While researchers have spent significant resources to address this challenge, most solutions impose significant energy, delay, and complexity overhead on energy-constrained IoT devices. In this paper, we first provide an overview of some of the techniques used to incorporate security and trust features into IoT devices. Then, we propose and demonstrate an innovative encryption approach for wireless IoT communications which is low-energy, low-complexity, and low-latency. The proposed cryptography integrates the encryption into the RF front-end of a wireless transceiver and is energy-efficient, making it suitable for real-time and energy-limited IoT connectivity applications.

Index Terms—Hardware Security, Analog Security, Trusted Communication, Software-defined-radio, SDR, True-random-number-generator, TRNG

I. INTRODUCTION

With the ever growing desire to increase efficiency and functionality in industry, real-time device connectivity is becoming a necessity. Over the past decade, IoT has emerged as the industry chosen network platform for wireless device connectivity due to simple protocols and ease of hardware implementation [1, 2, 3]. The widespread deployment of connected IoT devices in industrial and consumer applications calls for ensuring the security and trust in the wireless communication link. However, the limited available energy in most IoT devices, such as those relying on energy harvested from the environment, complicates this task [4].

Security challenges have been found in different layers of the IoT network [5, 6]. To solve these challenges, conventional security protocols exploit software or hardware encryption to secure the data and defend against attacks [7]. As one of these two types, software encryption heavily relies on the processing power available in the network. It is commonly used in computers with powerful central processing units (CPUs). By integrating the encryption algorithms in the software and running the software on processors, custom modification to the hardware is avoided, reducing the complexity of the system.

Hardware encryption, on the contrary, implements cryptographic functions in dedicated circuitry [8]. This way, high

This work is supported by NSF CCSS Award #2029407 and partially supported by the Air Force Center of Excellence for Enabling Cyber Defense in Analog and Mixed Signal Domain (CYAN).

energy CPUs can be replaced with low energy Application-Specific Integrated Circuits (ASICs), relaxing the overall energy consumption in the IoT device. The system latency/number of cycles is also significantly reduced due to the elimination of time-consuming software operations, making these solutions more attractive for IoT applications [9].

Standard hardware security protocols employ modern digital encryption methods such as the Advanced Encryption Standard (AES) and Elliptic Curve Cryptography (ECC) to protect the data [4]. These algorithms can be broadly divided into two classes: symmetric cryptography or private-key encryption, and asymmetric cryptography or public-key encryption [10, Ch. 1, pp. 4] [11]. Although these solutions offer full data security, the amount of energy required for bit-by-bit encryption is prohibitively high for energy-constrained IoT devices [12]. The added delay in the data transmission is also significant, negatively affecting the latency of the link which is critical for many mission-critical and time-sensitive applications. To alleviate these problems, we propose an innovative low-energy hardware encryption solution in which the encryption and decryption is performed in the analog domain within the wireless transceiver, with minimal energy and delay overhead to the system. An overview of the proposed security implementation is shown in Fig. 1.

In the following section, we will provide an overview of state-of-the-art hardware security techniques in IoT applications with emphasis on hardware implementation of cryptography. Our proposed analog-inspired low-energy hardware security solution for IoT trusted communications is detailed in section III. Section IV provides concluding remarks.

II. PRIOR ART IN IoT DATA SECURITY

Due to the low-energy nature of IoT devices, they usually rely on hardware encryption to reduce the energy required to perform security operations. While most devices use standardized modern cryptographic algorithms such as AES, Rivest–Shamir–Adleman (RSA), or various forms of ECC, as can be seen in [13, 14, 15, 16], there has been an increasing desire to develop new methods specifically targeting embedded devices. This has resulted in the National Institute for Standards and Technology (NIST) organizing a competition to produce a lightweight cryptography standard for IoT devices [17]. Other researchers have developed methods that rely on

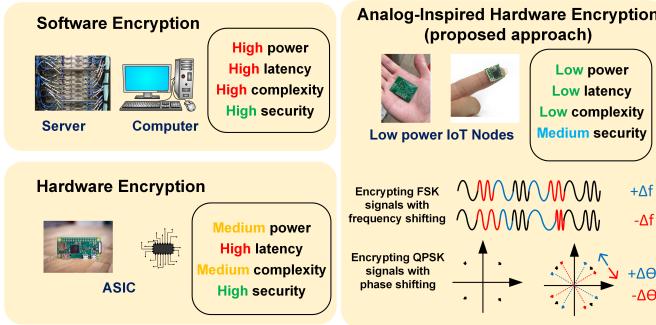


Fig. 1: Comparison between existing encryption approaches and the proposed approach.

the physical properties of the IoT device or the communication link in order to encrypt the data [18, 19, 20, 21, 22]. In general, all of these hardware encryption methods can be classified into one of two types, symmetric encryption and asymmetric encryption. Both of these forms of encryption can be utilized in trusted IoT communications; therefore, we will briefly introduce them and highlight their differences.

A. Symmetric vs. Asymmetric Cryptography

Symmetric encryption utilizes a shared key between parties for both encryption and decryption. In contrast, asymmetric encryption uses two keys, a public-key for encryption and a separate private-key for decryption [10, Ch. 1, pp. 5]. A message encrypted by the public-key can only be read with the private decryption key (Fig. 2). However, asymmetric cryptography is significantly slower than symmetric cryptography; taking two to three orders of magnitude longer to perform than a similar private-key algorithm [10, Ch. 11, pp. 377]. Despite this limitation, public-key cryptography is widely used and complementary to private-key cryptography. Because asymmetric cryptographic algorithms can ensure the security of the message without a shared key, it is used for key transport between two parties as the first step during communication. After securely exchanging the key, symmetric encryption is used to encrypt the subsequent data stream. By combining asymmetric and symmetric encryption in a communication protocol, both key and data can be secured during data transmission [10, Ch. 11, pp. 389-399] [11]. In the next subsection, various implementations of these methods will be explored.

B. AES, RSA, and ECC Encryption

One of the most widely used symmetric cryptographic algorithm is the AES [23]. Given its ubiquitous utilization, there are many ASIC implementations of AES in literature, with varying focuses on either performance or energy efficiency [24, 25, 26, 27, 28, 29, 13]. Current AES implementations designed for low energy have a peak energy efficiency of around 340 Gbps/W corresponding to an energy use of about 10 pJ per bit [27, 13, 29, 30]. Note that the energy efficiency varies depending on the process technology used and encryption/decryption speed. Broadly, there are two approaches for

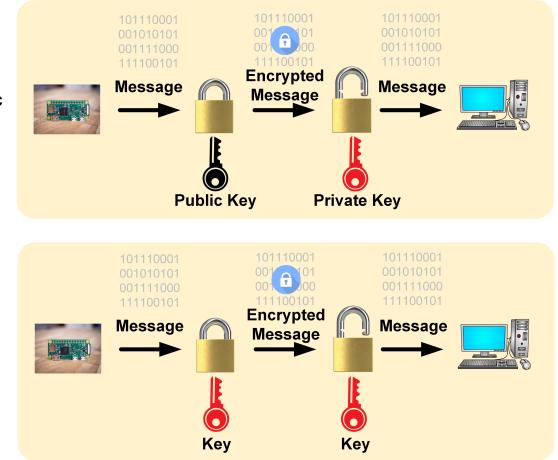


Fig. 2: Comparison between asymmetric and symmetric encryption.

hardware implementations of the AES algorithm, a highly pipelined approach for critical performance applications as demonstrated in [28], or of interest to IoT applications, area and energy optimized implementations that trade throughput for size and energy efficiency [27, 29, 13]. However, as noted in [31], many of these designs are susceptible to side-channel attacks; allowing an attacker to extract secret keys by probing power supply lines or scanning the integrated circuit's electromagnetic (EM) leakage [32]. This has led ASIC designers to modify their designs to mitigate these risks [27, 13]. Unfortunately, the mitigations increase the energy and area required to implement the AES circuit. For example, [27] reports a 28% increase in area and a 23% increase in power. Therefore, despite various advantages of AES encryption, it has serious drawbacks for low-energy IoT devices.

In order to securely transport the symmetric key between devices, some form of asymmetric encryption, such as RSA or ECC, is performed. Both of these algorithms have been implemented in ASICs and FPGAs. RSA implementations are seen in [33], [34], and [35] and ECC implementations in [36], [37], [38], and [30]. Compared to RSA, ECC can shorten the key length by $\approx 10x$ with the same level of security. However, RSA can achieve a shorter encrypting and decrypting time as compared to ECC [39]. In [9], an implementation of a full ECC accelerator is described. In this design, $0.2 \mu J$ per bit of energy is used to perform an encryption operation. This performance is comparable to hardware implementations of other, similar asymmetric algorithms [30, 40, 41]. This demonstrates the significant energy overhead of ECC or RSA encryption over AES encryption, clearly showing why it is only used for initial key exchange. The high energy usage of ECC and RSA algorithms makes them unacceptable for low energy IoT devices [42].

C. Low-energy Digital Cryptography

In order to better enable cryptography in energy constrained devices, NIST initiated the development of a new set of

symmetric algorithms specifically tailored for low energy applications [17]. In March of 2021, ten finalist algorithms were selected and are currently under public review [43]. In addition to targeting low energy, these algorithms aim to minimize the possibility of leaking information through side channels, such as power and EM leakage, as has been an issue with implementations of previous algorithms [17]. One of the finalist algorithms for NIST (Grain-128AEAD) was evaluated in [44]. In the paper, researchers built two implementations of the algorithm targeting high throughput and low energy respectively. The resulting energy usage from each implementation was 4.4 pJ per bit when targeting high throughput and 0.3 pJ per bit when targeting low energy operation. As a comparison, implementations of the AES algorithm have an energy usage of 10 pJ per bit as documented in [27, 13, 29, 30]. Other hardware implementations of the NIST algorithms are documented in [45], with benchmarks for energy, area and throughput included. While the NIST lightweight cryptography algorithms focus on symmetric encryption, there have also been recent developments in low-energy asymmetric encryption suitable for IoT devices [46]. In [9], the authors built an IC based on the Ring-Learning-With-Errors Key-Exchange algorithm (RLWE-KEX). This is a lattice-based, asymmetric cryptography algorithm, designed to be secure against quantum computers. In their implementation, the researchers achieved an efficiency of 3.4 nJ per bit. This is 30x more energy-efficient compared to ECC algorithms with similar security [30]. While these developments show impressive increases in energy efficiency over standard algorithms, they still are a significant burden for low-energy IoT devices.

D. Physical Encryption Methods

Another approach to reduce the energy required for asymmetric encryption is to leverage physical differences that occur among chips due to process variations. In [18], utilizing Physical Unclonable Functions (PUFs) for public-key cryptography was analyzed. Using the delay of logic networks as a PUF (XOR network [20], NAND and XOR array [47]), one can achieve asymmetric encryption and take advantage of the process-variation-induced random delay to secure the data. This provides a novel alternative to traditional public-key cryptographic methods, with lower energy use [19]. However, PUFs, in general, suffer from added complexity and stability issues.

A corollary to this, but for symmetric encryption, is to apply analog encryption at the modulation stage of the system. This security mechanism builds off the foundational work of Wyner in [48], which describes conditions under which parties can communicate in secret while in the presence of an eavesdropper without sharing cryptographic keys. The limitation of [48] is that it requires the intended receiver to have a higher channel capacity than the eavesdropper. A method to circumvent this limitation is to purposefully distort the transmitted signal in a way that is known to the intended recipient, but hidden from an eavesdropper. In this way, the channel, as described by Wyner, can be employed.

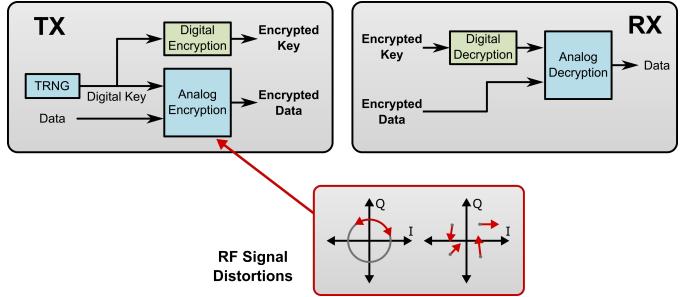


Fig. 3: Proposed analog-based encryption system.

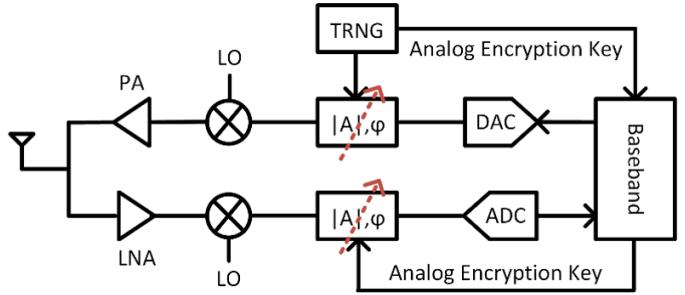


Fig. 4: Analog encryption transceiver – Phase shifting and TRNG are added to the RF front-end.

This technique has been successfully demonstrated in optical communication systems [49, 50]. This symmetric encryption method can be combined with the unique characteristics of an individual RF transmitter to perform key exchange allowing for low energy, secure communication [4, 51, 19, 52].

Adding security features at the modulation stage can be beneficial for IoT devices with simple transceivers and low energy budget. In the next section, we introduce our proposed analog-based encryption for low energy IoT device communication. By integrating the encryption process into the modulation stage, we aim to achieve low-latency and low energy operation in IoT devices, while also supporting real-time operation.

III. ANALOG ENCRYPTION

A. High Level Overview

Our solution to the problem of high-energy bit-by-bit digital encryption is a form of analog encryption that takes place inside the RF frontend of the radio. In doing so, it removes the need for digital encryption for much of the transmitted data, reducing the energy requirement for the system. Analog encryption is performed by distorting the transmitted waveform inside the RF frontend. Due to the distortion, an eavesdropping receiver will experience a highly elevated bit-error-rate (BER) when trying to demodulate the data, preventing it from correctly reading the information. The goal of the system is to keep an eavesdropper's BER close to 0.5, as this deprives them of all information about the transmitted data. This method is particularly well suited for applications where the goal is temporal security, where information needs to be protected for a few hours or days before it goes stale to an attacker. This

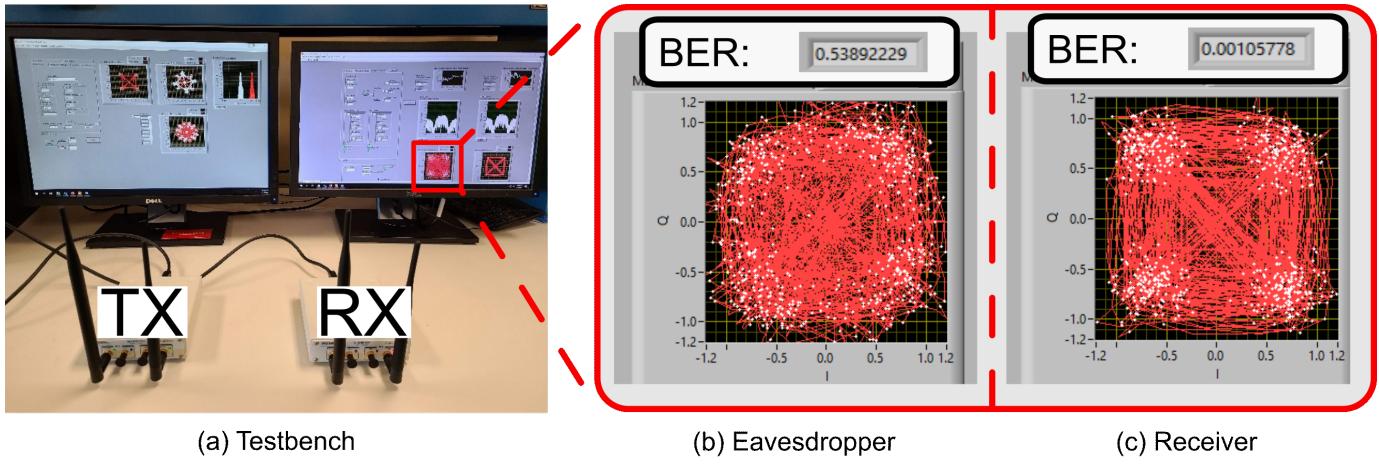


Fig. 5: Software defined radio implementation: (a) shows the testbench of the system, (b) is the constellation of an eavesdropping receiver (BER of 0.538), and (c) shows the constellation diagram of an intended receiver (BER of 0.001).

contrasts with absolute security which requires thousands of years of computing time to decrypt.

Fig. 3 shows the the basic process of analog encryption. Distortion is applied at the transmitter based on a one-time-use digital-key. The distortion can take one of many forms such as shifting the phase, amplitude, or frequency of the transmitted waveform. In this work, we focus on shifting the signal phase as it is most applicable in IoT devices using low-order modulation schemes, such as QPSK (Quadrature Phase Shift Keying). The key used to apply the distortion is produced by a true-random-number-generator (TRNG) which is then digitally encrypted and transmitted to the receiver (Fig. 3). Since the key is determined by random noise outside the control of an eavesdropper, the eavesdropper has no information about the value of the key. In contrast, since the intended receiver has access to the key, it can invert the distortion applied to the signal and correctly demodulate the data.

A high-level view of how the system is implemented is shown in Fig. 4. This shows that the analog encryption method requires the addition of a high resolution phase shifter in the transmitter and receiver, and a TRNG at the transmitter to produce digital-keys. This removes the need for bit-by-bit encryption of the transmitted data at both the transmitter and receiver, and hence have the potential to significantly reduce the energy consumption and latency. Note that the digital-key still requires traditional encryption; however, it is much shorter than the overall data packet. Therefore, encrypting the digital-key requires less energy than digitally encrypting the entire payload data. For increased security, the key can be regenerated and re-transmitted to the receiver. This will prevent an eavesdropper from recovering the key after its use for some time. Frequent updates of the key may also help mitigate side channel leakage of information. The refresh rate of the key can be varied by the system for various operating environment. In an ASIC implementation of the system, care is required when designing the circuitry necessary to implement the analog encryption (phase shifter and TRNG), particularly

in the area of energy use, as these are the main contributors to the system overhead.

B. Software Defined Radio Implementation

To test the efficacy of our analog encryption system, a software-defined-radio (SDR) based implementation is utilized. The Software-defined-radio allows for the direct control of the modulation and demodulation of a radio signal by streaming the raw in-phase (I) and quadrature (Q) data between the transceiver and a host computer. As there is direct control over the radio front-end by software, it allows for the rapid development of new communication protocols. Unfortunately, due to the general-purpose nature of the SDR hardware, it does not provide a comparable level of energy efficiency that a dedicated application specific RF front-end does. Also, due to most of the implementation being performed in software on a desktop computer, a strong one-to-one correlation between changes to the system and overall energy use is not feasible; thereby making power measurements difficult to impossible to perform accurately. However, while the SDR implementation is unable to give results for system power use, it provides a means to evaluate the fundamental operation of the analog encryption scheme. We leverage this capacity to test the ability of our system to perform reliable communication and substantially increase the BER of an eavesdropper.

Our experimental setup consists of two SDRs. One acting as a transmitter, the other as a receiver (Fig.5a). The transmitting SDR performs the analog encryption on the transmitted data packets and the receiver performs the inverse operation to obtain the original data. Analog encryption can be enabled or disabled by enabling or disabling the TRNG used to produce digital-keys; while the number of bits used to perform the phase shifting can also be adjusted. This allows for testing the effect of phase shift resolution on the eavesdropper BER. These keys are transmitted over a second channel so that they can be used by the receiver. At the receiver, the decoding of the digital-key can be disabled; allowing the receiver to emulate

the experience of an eavesdropping receiver. By comparing the data transmitted to the data extracted at the receiver, the number of incorrect bits are found and used to calculate the BER of the system.

C. Results

BER is used as a metric to determine the validity of our transceiver and analog cryptography implementation. Our goal for the system is that an eavesdropper experiences a BER of about 0.5, so that decoding the message is impossible; however, the intended receiver should have a very low BER. With our testbench, the BER for the simulated eavesdropping receiver stayed near 0.5 within a range of ± 0.13 as long as more than two bits were used for the random phase shifting (Fig. 5b). When using more than 2-bits of phase shifting resolution, the BER was not significantly impacted and stayed close to 0.5 for the eavesdropper. When operating as the intended receiver, the BER fell to less than 0.005 (Fig. 5c). This is the same BER achieved by a reference QPSK transceiver design implemented on the same SDR hardware without exercising any encryption. The SDR implementation of this system demonstrates a wireless dual channel approach for encrypting QPSK modulated signals, allowing the main channel payload to be randomly phase shifted, while the second channel carries the value of the random phase shift for decryption by the receiver.

IV. CONCLUSION

This paper provides an overview of existing solutions for IoT device security. Current digital encryption technologies have limitations for IoT devices, leading to the development of new low-energy solutions. While these new methods include lightweight digital encryption algorithms, there has also been promising research to develop physical approaches to enable IoT security. We propose one such method that uses analog encryption methods embedded in the RF front-end to protect a wireless communication link. This method promises a low-energy and low-latency alternative to bit-by-bit digital encryption methods. We also demonstrate a proof-of-concept implementation of our technique using an SDR platform. This uses phase shifting to increase the BER of an eavesdropping receiver to prevent it from correctly decoding transmitted information.

REFERENCES

- [1] S. Madakam, R. Ramaswamy, and S. Tripathi, "Internet of Things (IoT): A Literature Review," *JCC*, vol. 03, no. 05, pp. 164–173, 2015. [Online]. Available: <http://www.scirp.org/journal/doi.aspx?DOI=10.4236/jcc.2015.35021>
- [2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167739X13000241>
- [3] I. Lee and K. Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises," *Business Horizons*, vol. 58, no. 4, pp. 431–440, Jul. 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0007681315000373>
- [4] J. Zhang, S. Rajendran, Z. Sun, R. Woods, and L. Hanzo, "Physical Layer Security for the Internet of Things: Authentication and Key Generation," *IEEE Wireless Communications*, vol. 26, no. 5, pp. 92–98, Oct. 2019, conference Name: IEEE Wireless Communications.
- [5] T. Rao and E. Haq, "Security Challenges Facing IoT Layers and its Protective Measures," *International Journal of Computer Applications*, vol. 179, pp. 31–35, Mar. 2018.
- [6] M. A. Latif, M. B. Ahmad, and M. K. Khan, "A Review on Key Management and Lightweight Cryptography for IoT," in *2020 Global Conference on Wireless and Optical Technologies (GCWOT)*, Oct. 2020, pp. 1–7.
- [7] M. Healy, T. Newe, and E. Lewis, "Analysis of Hardware Encryption Versus Software Encryption on Wireless Sensor Network Motes," in *Smart Sensors and Sensing Technology*, Jan. 2008, pp. 3–14.
- [8] T. Maude and D. Maude, "Hardware protection against software piracy," *Commun. ACM*, vol. 27, no. 9, pp. 950–959, Sep. 1984. [Online]. Available: <https://doi.org/10.1145/358234.358271>
- [9] U. Banerjee, A. Pathak, and A. P. Chandrakasan, "2.3 An Energy-Efficient Configurable Lattice Cryptography Processor for the Quantum-Secure Internet of Things," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, Feb. 2019, pp. 46–48, iSSN: 2376-8606.
- [10] J. Katz and Y. Lindell, *Introduction to modern cryptography*, 2nd ed., ser. Chapman & hall/crc cryptography and network security series. Boca Raton: CRC Press/Taylor & Francis, 2015.
- [11] H. Suo, J. Wan, C. Zou, and J. Liu, "Security in the Internet of Things: A Review," in *2012 International Conference on Computer Science and Electronics Engineering*, vol. 3, Mar. 2012, pp. 648–651.
- [12] M. Bloch, O. Günlü, A. Yener, F. Oggier, H. V. Poor, L. Sankar, and R. F. Schaefer, "An Overview of Information-Theoretic Security and Privacy: Metrics, Limits and Applications," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 1, pp. 5–22, Mar. 2021, conference Name: IEEE Journal on Selected Areas in Information Theory.
- [13] A. Singh, M. Kar, S. Mathew, A. Rajan, V. De, and S. Mukhopadhyay, "25.3 A 128b AES Engine with Higher Resistance to Power and Electromagnetic Side-Channel Attacks Enabled by a Security-Aware Integrated All-Digital Low-Dropout Regulator," in *2019 IEEE International Solid-State Circuits Conference - (ISSCC)*, Feb. 2019, pp. 404–406, iSSN: 2376-8606.
- [14] T. Kudithi and R. Sakthivel, "High-performance ECC processor architecture design for IoT security applications," *J Supercomput*, vol. 75, no. 1, pp. 447–474, Jan. 2019. [Online]. Available: <https://doi.org/10.1007/s11227-018-02740-2>
- [15] M. Suárez-Albela, T. M. Fernández-Caramés, P. Fraga-Lamas, and L. Castedo, "A Practical Performance Comparison of ECC and RSA for Resource-Constrained IoT Devices," in *2018 Global Internet of Things Summit (GIoTS)*, Jun. 2018, pp. 1–6.
- [16] B. Hammı, A. Fayad, R. Khatoun, S. Zeadally, and Y. Begriche, "A Lightweight ECC-Based Authentication Scheme for Internet of Things (IoT)," *IEEE Systems Journal*, vol. 14, no. 3, pp. 3440–3450, Sep. 2020, conference Name: IEEE Systems Journal.
- [17] K. A. McKay, L. Bassham, M. S. Turan, and N. Mouha, "Report on lightweight cryptography," National Institute of Standards and Technology, Gaithersburg, MD, Tech. Rep. NIST IR 8114, Mar. 2017. [Online]. Available: <https://nvlpubs.nist.gov/nistpubs/ir/2017/NISTIR-8114.pdf>
- [18] T. Xu, J. B. Wendt, and M. Potkonjak, "Security of IoT systems: Design challenges and opportunities," in *2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov. 2014, pp. 417–423, iSSN: 1558-2434.
- [19] Q. Zhou, Y. He, K. Yang, and T. Chi, "12.3 Exploring PUF-Controlled PA Spectral Regrowth for Physical-Layer Identification of IoT Nodes," in *2021 IEEE International Solid- State Circuits Conference (ISSCC)*, vol. 64, Feb. 2021, pp. 204–206, iSSN: 2376-8606.
- [20] N. Beckmann and M. Potkonjak, "Hardware-Based Public-Key Cryptography with Public Physically Unclonable Functions," in *Information Hiding*, ser. Lecture Notes in Computer Science, S. Katzenbeisser and A.-R. Sadeghi, Eds. Berlin, Heidelberg: Springer, 2009, pp. 206–220.
- [21] S. Taneja, V. K. Rajanna, and M. Alioto, "36.1 Unified In-Memory Dynamic TRNG and Multi-Bit Static PUF Entropy Generation for Ubiquitous Hardware Security," in *2021 IEEE International Solid- State Circuits Conference (ISSCC)*, vol. 64, Feb. 2021, pp. 498–500, iSSN: 2376-8606.

[22] T. Xu, J. B. Wendt, and M. Potkonjak, "Digital bimodal function: An ultra-low energy security primitive," in *International Symposium on Low Power Electronics and Design (ISLPED)*, Sep. 2013, pp. 292–296.

[23] J. Daemen and V. Rijmen, "Aes proposal: Rijndael," 1999.

[24] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, "Strong authentication for RFID systems using the AES algorithm," in *International workshop on cryptographic hardware and embedded systems*. Springer, 2004, pp. 357–370.

[25] M. Feldhofer, J. Wolkerstorfer, and V. Rijmen, "Aes implementation on a grain of sand," *IEE Proceedings-Information Security*, vol. 152, no. 1, pp. 13–20, 2005.

[26] N. Pramstaller, S. Mangard, S. Dominikus, and J. Wolkerstorfer, "Efficient aes implementations on asics and fpgas," in *International Conference on Advanced Encryption Standard*. Springer, 2004, pp. 98–112.

[27] R. Kumar, V. Suresh, M. Kar, S. Satpathy, M. A. Anders, H. Kaul, A. Agarwal, S. Hsu, G. K. Chen, R. K. Krishnamurthy, V. De, and S. K. Mathew, "A 4900- μ m² 839-Mb/s Side-Channel Attack-Resistant AES-128 in 14-nm CMOS With Heterogeneous Sboxes, Linear Masked MixColumns, and Dual-Rail Key Addition," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 4, pp. 945–955, Apr. 2020, conference Name: IEEE Journal of Solid-State Circuits.

[28] B. Erbagci, N. E. C. Akkaya, C. Teegarden, and K. Mai, "A 275 Gbps AES encryption accelerator using ROM-based S-boxes in 65nm," in *2015 IEEE Custom Integrated Circuits Conference (CICC)*, Sep. 2015, pp. 1–4.

[29] S. Mathew, S. Satpathy, V. Suresh, M. Anders, H. Kaul, A. Agarwal, S. Hsu, G. Chen, and R. Krishnamurthy, "340 mV-1.1 V, 289 Gbps/W, 2090-Gate NanoAES Hardware Accelerator With Area-Optimized Encrypt/Decrypt GF(2⁴)² Polynomials in 22 nm Tri-Gate CMOS," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 4, pp. 1048–1058, Apr. 2015, conference Name: IEEE Journal of Solid-State Circuits.

[30] U. Banerjee, A. Wright, C. Jivekar, M. Waller, Arvind, and A. P. Chandrakasan, "An Energy-Efficient Reconfigurable DTLS Cryptographic Engine for Securing Internet-of-Things Applications," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 8, pp. 2339–2352, Aug. 2019, conference Name: IEEE Journal of Solid-State Circuits.

[31] Y. Lu, K. Boey, M. O'Neill, and J. McCanny, "Practical comparison of differential power analysis techniques on an ASIC implementation of the AES algorithm," in *IET Irish Signals and Systems Conference (ISSC 2009)*, Jun. 2009, pp. 1–6.

[32] G.-l. Ding, Z.-x. Li, X.-l. Chang, and Q. Zhao, "Differential Electromagnetic Analysis on AES Cryptographic System," in *2009 Second Pacific-Asia Conference on Web Mining and Web-based Application*, Jun. 2009, pp. 120–123.

[33] A. Fournaris and O. Koufopavlou, "A new RSA encryption architecture and hardware implementation based on optimized Montgomery multiplication," in *2005 IEEE International Symposium on Circuits and Systems*, 2005, pp. 4645–4648 Vol. 5.

[34] A. Daly and W. Marnane, "Efficient architectures for implementing montgomery modular multiplication and rsa modular exponentiation on reconfigurable logic," in *Proceedings of the 2002 ACM/SIGDA tenth international symposium on Field-programmable gate arrays*, 2002, pp. 40–49.

[35] A. Mazzeo, L. Romano, G. P. Saggese, and N. Mazzocca, "Fpga-based implementation of a serial rsa processor," in *2003 Design, Automation and Test in Europe Conference and Exhibition*. IEEE, 2003, pp. 582–587.

[36] T. Kudithi and R. Sakthivel, "High-performance ecc processor architecture design for iot security applications," *The Journal of Supercomputing*, vol. 75, no. 1, pp. 447–474, 2019.

[37] C. H. Kim, S. Kwon, and C. P. Hong, "Fpga implementation of high performance elliptic curve cryptographic processor over gf (2163)," *Journal of Systems Architecture*, vol. 54, no. 10, pp. 893–900, 2008.

[38] K. Jarvinen and J. Skytta, "On parallelization of high-speed processors for elliptic curve cryptography," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 9, pp. 1162–1175, 2008.

[39] M. El-Haiii, M. Chamoun, A. Fadlallah, and A. Serhrouchni, "Analysis of Cryptographic Algorithms on IoT Hardware platforms," in *2018 2nd Cyber Security in Networking Conference (CSNet)*, Oct. 2018, pp. 1–5.

[40] P. Pessl and M. Hutter, "Curved Tags – A Low-Resource ECDSA Implementation Tailored for RFID," in *Radio Frequency Identification: Security and Privacy Issues*, N. Saxena and A.-R. Sadeghi, Eds. Cham: Springer International Publishing, 2014, vol. 8651, pp. 156–172, series Title: Lecture Notes in Computer Science. [Online]. Available: http://link.springer.com/10.1007/978-3-319-13066-8_10

[41] M. Hutter, J. Schilling, P. Schwabe, and W. Wieser, "NaCl's Crypto_box in Hardware," vol. 9293, Sep. 2015, pp. 81–101.

[42] M. A. Bahnsawi, K. Ibrahim, A. Mohamed, M. K. Mohamed, A. Moustafa, K. Abdelmonem, Y. Ismail, and H. Mostafa, "ASIC-oriented comparative review of hardware security algorithms for internet of things applications," in *2016 28th International Conference on Microelectronics (ICM)*, Dec. 2016, pp. 285–288.

[43] I. T. L. Computer Security Division, "Lightweight Cryptography | CSRC | CSRC," Jan. 2017. [Online]. Available: <https://csrc.nist.gov/projects/lightweight-cryptography>

[44] J. Sönnnerup, M. Hell, M. Sönnnerup, and R. Khattar, "Efficient Hardware Implementations of Grain-128AEAD," in *Progress in Cryptology – INDOCRYPT 2019*, F. Hao, S. Ruj, and S. Sen Gupta, Eds. Cham: Springer International Publishing, 2019, vol. 11898, pp. 495–513, series Title: Lecture Notes in Computer Science. [Online]. Available: http://link.springer.com/10.1007/978-3-030-35423-7_25

[45] M. D. Aagaard and N. Zidaric, "Asic benchmarking of round 2 candidates in the nist lightweight cryptography standardization process," *Cryptology ePrint Archive*, Report 2021/049, 2021, <https://ia.cr/2021/049>.

[46] I. Verbauwhede, J. Balasch, S. S. Roy, and A. Van Herrewege, "24.1 Circuit challenges from cryptography," in *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*, Feb. 2015, pp. 1–2, iSSN: 2376-8606.

[47] M. Potkonjak, S. Meguerdichian, A. Nahapetian, and S. Wei, "Differential public physically unclonable functions: Architecture and applications," in *2011 48th ACM/EDAC/IEEE Design Automation Conference (DAC)*, Jun. 2011, pp. 242–247, iSSN: 85-644924.

[48] A. D. Wyner, "The wire-tap channel," *The Bell System Technical Journal*, vol. 54, no. 8, pp. 1355–1387, Oct. 1975, conference Name: The Bell System Technical Journal.

[49] G. S. Kanter, D. Reilly, and N. Smith, "Practical physical-layer encryption: The marriage of optical noise with traditional cryptography," *IEEE Communications Magazine*, vol. 47, no. 11, pp. 74–81, Nov. 2009, conference Name: IEEE Communications Magazine.

[50] M. Yoshida, T. Hirooka, K. Kasai, and M. Nakazawa, "QAM quantum noise stream cipher using digital coherent optical transmission," in *2015 49th Asilomar Conference on Signals, Systems and Computers*, Nov. 2015, pp. 1007–1011, iSSN: 1058-6393.

[51] C. T. Zenger, M. Pietersz, J. Zimmer, J.-F. Posielek, T. Lenze, and C. Paar, "Authenticated key establishment for low-resource devices exploiting correlated random channels," *Computer Networks*, vol. 109, pp. 105–123, Nov. 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1389128616301943>

[52] C. Zhang, J. Yue, L. Jiao, J. Shi, and S. Wang, "A Novel Physical Layer Encryption Algorithm for LoRa," *IEEE Communications Letters*, pp. 1–1, 2021, conference Name: IEEE Communications Letters.