SHA-512 Based Wireless Authentication Scheme for Smart Grid Battery Management Systems

Ahmad Al Khas*, Ihsan Cicek*‡

* Integrated Circuits Laboratory (ICLAB), Department of Electrical-Electronics Engineering, İstanbul Şehir University

(ahmadkhas@std.sehir.edu.tr, ihsancicek@sehir.edu.tr)

‡ Corresponding Author; Ihsan Cicek, Orhantepe Mahallesi, Turgut Ozal Bulvarı,

No: 21, 34865 Dragos, Kartal/Istanbul, Turkey, Tel: +90 444 40 34 - 9420,

Fax: +90 216 474 53 53, ihsancicek@sehir.edu.tr

Received: 12.17.2020 Accepted:04.02.2020

Abstract- Modern cyber-physical systems depend on the battery backup systems for continuous service. Due to excessive wiring requirements, the traditional methods used for battery authentication cannot be applied to modern smart batteries, especially when they are composed of large number of cells. In this work, we proposed a wireless battery authentication scheme for use with the battery management system and the cells to prevent potential hardware attacks through the trojan cells. We designed a SHA-512 IP Core which operates at 157 MHz and occupies 974 CLB slices and one block RAM on a Xilinx Artix-7 FPGA device. We integrated the SHA-516 module with a synthesizable CPU to be able to create a real-world test scenario and finally, we verified the correct operation of an example authentication protocol through a wireless communication channel established with the use of two ESP8266 based Wi-Fi modules.

Keywords SHA-512; FPGA; Authentication; Microblaze; Battery Management System; Smart grid.

1. Introduction

The dependency on inefficient manual processes for remote energy management have been evolving towards automated and smart cyber-physical systems that can be efficiently monitored and controlled. This paradigm shift has led to the development of new energy management systems that become the most crucial component of energy distribution in a large spectrum of applications from smart cities to electrical vehicles many of which depend on batteries as the power source [1]–[3]. As a result, the use of a battery management system (BMS) as an integrated cyber-physical module provides the distinct advantages of increased system reliability and availability, however this also raised security concerns [4]–[6]. Smart battery systems employed in electric vehicles or smart-grid backup systems are usually manufactured using numerous battery cells, which are connected to provide energy to the host system through the BMS [7]. The ignorance of hardware security in the design of such systems could easily lead to a catastrophe for the cyber physical system if a rogue battery cell as a hardware trojan is inserted into the battery pack in the factory manufacturing or in field maintenance [8]. This Trojan battery can deceive the BMS and its managing policies for energy by providing corrupted status for health. In addition, it can also act like a low resistance load and instead of providing energy, it can consume the available energy to overheat and cause a fire. These potential intrusive attacks can be mitigated if cryptographic techniques are used to authenticate the cells for genuineness before starting the system and during the operation of the battery pack.

In fact, there have been several products commercialized on the market which adapted the battery pack authentication scheme, especially for adapters used to power up portable computers and many other household goods [6], [9]. One of the popular algorithms is KEELOQ which is developed by Microchip Technology Inc. was broken as a result of its architectural weaknesses [10], [11]. Another one that is used in battery authentication is based on the eXtended Tiny Encryption Algorithm (XTEA) forked from publicly available Tiny Encryption Algorithm (TEA) [12]. Texas Instruments Inc., a well-known US semiconductor manufacturer, provides propriety commercial schemes for battery authentication and security [13]. However, these commercial solutions have been developed to authenticate

battery packs not individual cells and in typical scenarios the battery packs in many portable devices has limited number of cells. Unfortunately, these commercial solutions are not scalable by design, and cannot be used with the batteries consisted of large number of cells. Additionally, in many of the commercial solutions, there is a need for physical contacts with the batteries. When battery packs with large number of cells are considered for authentication using these methods, the need for physical connections to the cells render not only the design and manufacturing, but also the authentication infeasible. The additional wiring required for these physical contacts becomes the limiting factor for an economic and efficient smart battery implementation due to increased cost and hardware complexity. In our humble opinion, this problem can be addressed, and a scalable solution can be built if wireless communication techniques are utilized for the authentication of cells. This approach also makes the wireless health monitoring possible [14]. Each battery cell with a unique hardware identification code can be authenticated one by one to ensure the security while the monitoring capability would improve the safety of the battery. State-of-the-art low power wireless integrated circuit technologies, cryptographic accelerators, and synthesizable soft processors can be combined to design and fabricate special integrated circuits as a single chip solution.

In this study, we propose a wireless authentication method for use in batteries that are built using a large number of cells that are infeasible to get authenticated using traditional solutions that are based on physical wiring. We designed a working conceptual model in hardware using Field Programmable Gate Arrays (FPGAs) and we have verified the correct operation of our design in both simulation and in practical application. The outline of our contributions can be listed as the following:

1- We have developed an authentication method based on the second-generation Secure Hash Algorithms (SHA-512) for remotely verifying the genuineness of the cells. The main distinctive property of our approach is the simplicity and the scalability when compared with the existing traditional approaches.

2- In order to validate the proposed concept, we have developed an IP core for SHA-512 first, then we integrated it with a softcore processor that is synthesizable on FPGA devices. Two FPGA development boards were wirelessly connected to each other using commercial Wi-Fi modules for the emulation of remote communications.

3- For evaluating our design, we have built a communication channel that involves a server that emulates the BMS and a client in the role of a battery cell to be authenticated by the BMS through the wireless communication channel established.

The details of the authentication scheme are explained in the second Section. We have provided the details on the design and customization of the FPGA IP core along with its integration with the softcore processor in Section 3. Finally, in Section 4, we presented the correct operation of our design in practice.

2. Authentication Scheme Based on SHA-512

SHA-512 is an algorithm listed under secure hash algorithms, which are directional one- way functions that are free of any collisions. Usually these algorithms are used for generating unique digested data for every corresponding input. Being a collision-free function enables a one-to-one uniqueness and an unparalleled output data representation for each input data, even if the difference between the inputs is a one bit, the resulting output will be different. This property makes the authentication projects possible in practice [15]. In addition, these functions also enable the secure networking applications that are based on data integrity and authentication [16]. After the invalidation of the first generation of Secure Hash Algorithm SHA-1 by successful attacks [17]. The National Institute of Standards and Technology (NIST) has published the second-generation SHA-2 as a replacement. SHA-2 family of algorithms has been very popular in practice. Many information security applications such as TLS, SSL, SSH, PDP and IPSec have employed SHA-2 in their respective data processing steps. Although a more recent family of SHA has been added to the standards of NIST, known as the SHA-3 family [18]. The popularity of SHA-2 is still intact [19]. Moreover, SHA-3 is considered more complex in terms of hardware implementation, so we have decided to use SHA-512 of the second-generation family for its simplicity and ubiquity and larger size message digests.

Algorithmic flow of SHA-512 is quite simple and based on operations such as message padding, parsing and unidirectional hashing functions as presented in Fig. 1. The following three steps describe the pre-processing of SHA-512 briefly, while the computation of the hashed value is calculated by using the functions defined according to [19]:

Step 1: Message padding (Padded message should be consisted of a multiple of 1024-bits).

Step 2: Message parsing (The padded message is pared into N x 1024-bit blocks).

Step 3: Initializing the first hash value (contains eight separate 64-bit words in hexadecimal representation).

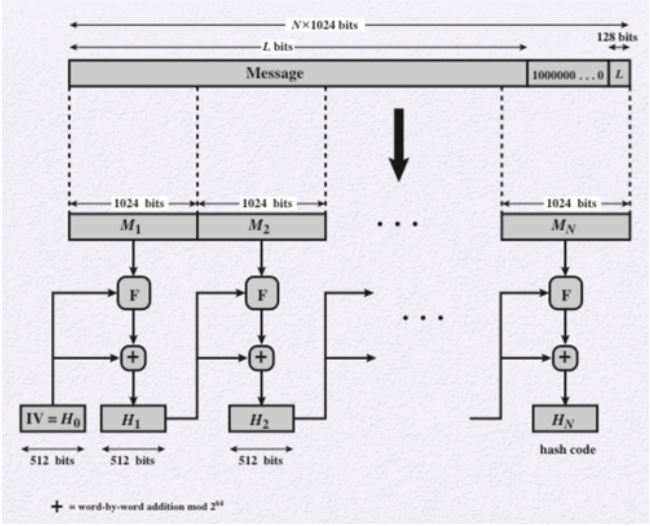

Fig. 1. The algorithmic flow of SHA-512 algorithm.

According to the algorithmic flow of the SHA-512 shown in Fig. 1, the message to be processed is partitioned into M-blocks each consisting of 1024 bits first, then the initial hash value H0 is set before the start of any operation on the blocks. At the end, the hash value of the message is computed after the processing of the last block, and the final calculated data is transferred to the output.

3. FPGA Hardware Implementation of the SHA-512 Algorithm

Reconfigurability along with acceleration capabilities of FPGA devices provide the opportunity for testing and predicting the correct operation of concept digital circuits before their fabrication in the form of application specific integrated circuits [20]. Consequently, we realized the SHA-512 base authentication scheme on an FPGA development board that hosts a Xilinx Artix 7 family device (XC7A35T-1CPG236C).

3.1. Design of the SHA-512 Hardware IP Core

SHA-512 is principally based on shuffling the input with some predefined constants, then calculating the hash values via corresponding specific functions defined for each step, then after the acquisition of all inputs, the final hash values are used to calculate the output digest value of the SHA-512. Fig. 2 presents top-level block diagram of our SHA-512 module, which accepts 64-bits as input and computes the digest message output as 512-bit data.

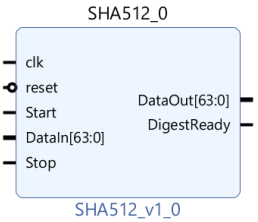

Fig. 2. SHA-512 module and interface signals.

Following the power on state, our design stays in idle after the bitstream file is loaded into the FPGA, and it stays in idle until the start command input is pulled high. Following that, the design starts to receive 64-bits of data words at its input on every rising edge of a clock operating at 78.5 MHz. After all the message is loaded in 64-bit blocks at a time, the stop command input is set to logic high after which the computation process starts operating at a clock frequency of 157 MHz. At the end of the calculation a digest ready flag signal is raised to logic high for a duration of eight 64-bit blocks to indicate the availability of the 512-bit final hash value at the output of the module. HDL Simulations of the proposed IP core design has been done to verify its functionality, and an example simulation outcome for our module is presented in Fig. 3. SHA-512 functional simulation results show that for the input data ”abcdabcd” in hexadecimal format, the computed hash output is obtained as 7edbb31279e6b88ac79812e2f77f5b234f817797c7cf98263d557ecfc992f1c43e8b169e11e3aaceb4407da8390517cac5e64f579344e15f589be5c20e7cecc8, which is equal to the expected value. As observed from the simulation waveform, it took around 142 µs for the digest output to be computed after the process had started. Studies in the state of the art achieved better results from this perspective. Reference [21] presented a SHA-512 implementation, which the digest message output for it was computed after approximately 55 µs. Our primary design constraint was area optimization to yield a compact and low-power module, so we traded the space with time.

Vivado design environment software provided by Xilinx Inc. was used to simulate and implement our SHA-512 design which was modelled using Verilog HDL. Table 1 shows our results for SHA-512 implementation along with other implementations from the literature. We have observed that our design is far superior to other existing designs in terms of resource utilization and maximum achievable clock frequency of 157 MHz, at the cost of latency for generating the digest output. We have also implemented an optimized version of the SHA-512 module on another Xilinx Artix 7 device (XC7A100T-CSG324C) using Digilent Nexys4 FPGA development board. The utilization of resources from the FPGA device is way more effective as well as the maximum frequency of the clock used is 157 MHz as observed in the table. Moreover, we employed another version that is an optimized one for our IP core design on a Xilinx Artix 7 (XC7A100T-CSG324C) device, by using Nexys4 FPGA development board. We have achieved a maximum clock speed of 189.5 MHz at the cost of 1307 CLB slices as shown in Table 1.

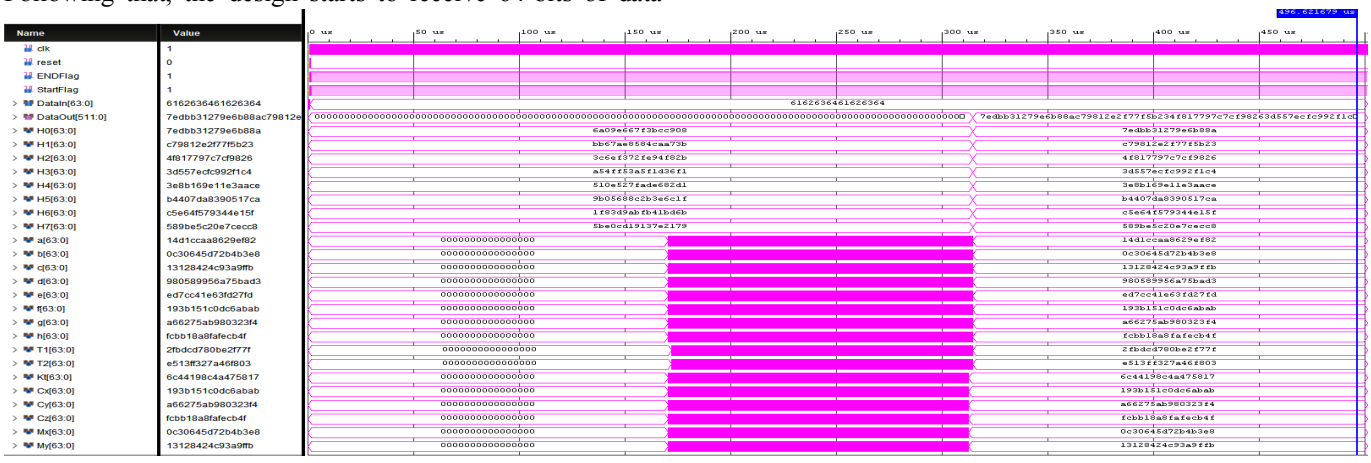

Fig. 3. SHA-512 simulation shows the correct operation of the designed IP Core.

Table 1. FPGA implementation results of SHA512 module with literature comparison.

	SHA-512 XC7A35T	SHA-512 XC7A100T	[22]	[23]	[24]	[25]	[26]	[27]
Max. Clock Freq. (MHz)	157	189.5	38	118.8	50	91.2	111.56	129
BRAMs	1	1	2	N/A	N/A	N/A	N/A	N/A
CLBs	974	1307	2914	2104	4957	7219	1613	1080
Max. Input (bits)	3584	3584	N/A	N/A	N/A	N/A	N/A	N/A

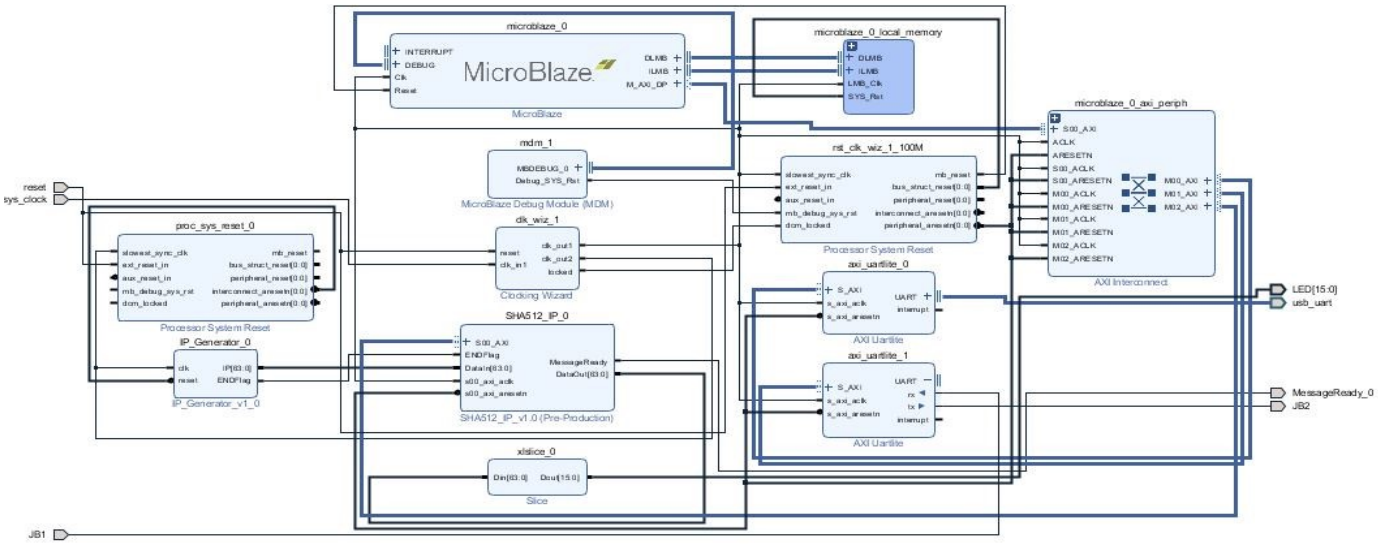

Fig. 4. The system level block diagram of the SHA-512 module integrated with the Micoblaze CPU.

3.2. Integration of the SHA-512 Module with A Synthesizable Processor

We have integrated our SHA-512 module with the synthesizable softcore processor, Microblaze, to build a working model for the battery cell authentication scenario. This approach allows computer monitoring of the SHA-512 core and facilitates data transfer while testing the wireless battery authentication scenario. We have designed a customized IP that encloses the SHA-512 module and establishes an interface with the AXI4 bus of the Microblaze CPU. As the MicroBlaze comes with many optional features, it had to be configured with the right specifications that would give a minimum footprint while providing the required functionality. As a result, a barebone CPU with no cache memory is used, and a local 16-KByte memory has been configured. The system clock used by the CPU is configured at 157 MHz, to be compatible with the operation of the SHA-512 module and to reduce overhead hardware clocking. Furthermore, Fig. 4 shows the entire hardware system design that includes the processor and SHA-512 module as well as other IPs like reset and clocking modules and AXI4 bridges integrated for a fully functional system-on-chip system. We have added two IP cores of UART-Lite for enabling the RS232 serial communication protocol with the outside environment at a baud rate of 115200 bps. We have used two UART-Lite IP cores which means there were two ports established, one is for debugging purposes using the computer while the other is connected to input-output pins on one of the PMOD ports on the FPGA development board to communicate with the ESP8266 based Wi-Fi module which is dedicated to establish wireless communication. There are two FPGA boards configured similarly communicating through the established channel, we have used two available ESP8266 based Wi-Fi modules. Meanwhile, the communication channel can be established using other wireless modules like ZigBEE, or Bluetooth. Each of the two FPGA boards has a role in our scenario. One board was the server acting as BMS and the other one emulated the smart cell of battery as a client. We have presented our FPGA hardware implementation results in Table 2, and we report an estimated power consumption of 354 mW for the system-on-chip.

Table 2. FPGA implementation results of the SHA512 module integrated with the CPU.

Hardware Specification	Parameter Value
Maximum Clock Frequency	157 MHz
Estimated Power Consumption	0.354 W
Number of BRAMs	20
Number of URAMs	0
Number of CLB slices	2217
Number of LUTs	8866
Number of FFs	9122
Number of DSPs	0

4. Testing and Performance Evaluation

We used the Xilinx software development kit (SDK) environment in order to run the test program written in C and verify the authentication process. Xilinx SDK can synthesize the required software libraries for the hardware configuration to ease the development of the application software. The wireless modules were pre-configured to set up a wireless communication channel using an IP protocol between them. One serial port handled uploading and debugging the software written in C, and the other one was used to handle the data transfer between FPGA and wireless modules. In Fig. 5, the process flow of authentication between battery cell and BMS is presented. We have used the output of the simulation presented in Fig. 3 in our hardware implementation as the unique ID, for validating the correct operation of SHA-512 in practical tests as shown in Fig. 6.

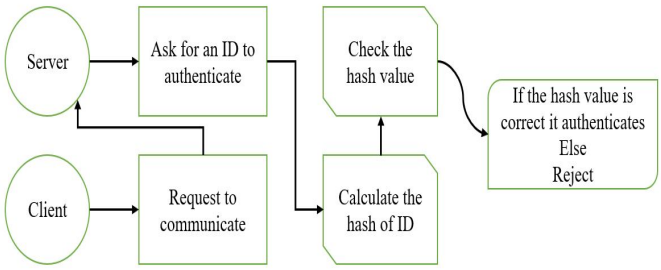

Fig. 5. SHA-512 based authentication protocol implemented in software.

In a real-world scenario, SHA-512 based wireless authentication protocol can be used to first identify the battery cells that will be remotely monitored, then the continuous health monitoring such as voltage, current, charge rate measurements can be accepted as genuine to protect the BMS against the intrusions using a trojan battery. When the authentication fails for a trojan or counterfeited battery cell, BMS can report it to the host system, and raise an alarm or disable the cell if it has some means of control. The correct operation of the authentication scheme was experimentally verified as shown by the terminal window in Fig. 7.

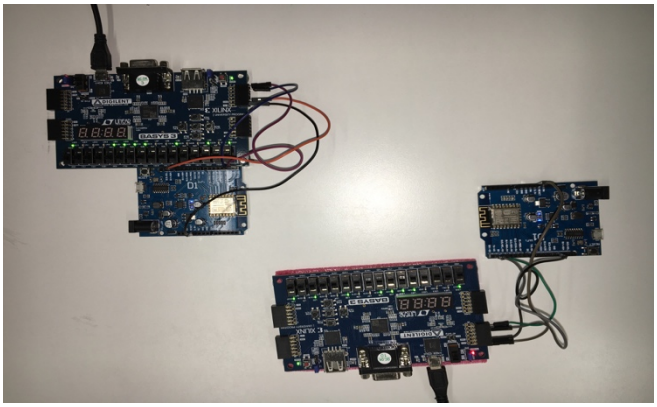

Fig. 6. SHA-512 based battery cell authentication test setup.

Fig. 7. Terminal window showing an example of the authentication operation.

5. Conclusion

In this study, we designed and implemented a SHA-512 based hardware module for use in a BMS-to-battery cell authentication scenario, which uses a wireless communication channel for creating a scalable solution. Proposed approach can be used for battery cell authentication to avoid counterfeited or hardware trojan battery cells. We designed and characterized a small footprint and fast SHA-512 IP core. Our design is better in terms of area and speed when compared to the state-of-the-art designs in the literature. The IP core occupies 974 CLB slices with the use of only one Block RAM on a Xilinx Artix-7 FPGA device, which can operate up to 157 MHz. We have integrated the designed IP core with a Microblaze synthesizable processor for the use in a wireless authentication scheme that we have proposed. We created a real-world test scenario using two ESP8266 Wi-Fi modules and two FPGA development boards that mimic the BMS and the battery cell. We have successfully verified the correct operation of the proposed wireless authentication scheme in this test setup.

References

[1] A. V. D. M. Kayem, S. D. Wolthusen, and C. Meinel, eds., Smart MicroGrid Systems Security and Privacy, vol. 71 of Advances in Information Security. Springer, 2018.

[2] A. Bani-ahmed, A. Nasiri, and I. Stamenkovic, “Foundational Support Systems of the Smart Grid: State of the Art and Future Trends,” International Journal of Smart Grid, ijSmartGrid, vol. 2, no. 1, pp. 1–12, 2018.

[3] W. Feng, C. Shuyu, L. Z. Kiat, C. Xuebing, and K. J. Tseng, “A NonInvasive On-line Condition Monitoring and Health Prognosis System for a Remote Islanded Micro-Grid,”6th IEEE International Conference on Smart Grid, icSmartGrids 2018, pp. 46–51, 2019.

[4] H. Rahimi-Eichi, U. Ojha, F. Baronti, and M. Chow, “Battery management system: An overview of its

application in the smart grid and electric vehicles," IEEE Industrial Electronics Magazine, vol. 7, pp. 4–16, June 2013.

[5] R. Hu, "Battery Management System For Electric Vehicle Applications," Master's thesis, University of Windsor, Ontario, Canada, 2011.

[6] M. Rezal, A. Zulaikha, M. Sabri, R. Yusof, and S. Ridzwan, "Orion battery management system (BMS) for lithium-ion battery pack," Colloquium of Education, Engineering & Technology 2014 (COLEET 2014), pp. 80–86, 2014.

[7] H. Abdelkader, A. Meriem, C. Ilhami, and K. Korhan, "Smart grid and renewable energy in Algeria," 6th International Conference on Renewable Energy Research and Applications, ICRERA 2017, pp. 1166–1171. 2017.

[8] M. Yesilbudak and I. Colak, "Main Barriers and Solution Proposals for Communication Networks and Information Security Smart Grids,"6th IEEE International Conference on Smart Grid, icSmartGrids 2018, pp. 58–63, 2019.

[9] K. Dietz, "Battery authentication for portable power supplies," Power Electronics Technology, vol. 32, no. 4, pp. 34–39, 2006.

[10] T.I. Inc, "Application Report: Battery Authentication and Security Schemes," no. August, pp.1–6, 2014. http://www.ti.com/lit/an/slua346a/slua346a.pdf. Last visited on 8 July 2019.

[11] T. Eisenbarth, T. Kasper, A. Moradi, C. Paar, M. Salmasizadeh, and M. T. M. Shalmani, "Physical cryptanalysis of keeloq code hopping applications." Cryptology ePrint Archive, Report 2008/058, 2008.

[12] T. Eisenbarth, T. Kasper, A. Moradi, C. Paar, M. Salmasizadeh, and M. T. M. Shalmani, "On the power of power analysis in the real world: A complete break of the keeloqcode hopping scheme," in CRYPTO, 2008.

[13] D. J. Wheeler and R. M. Needham, "Tea, a tiny encryption algorithm," in Fast Software Encryption (B. Preneel, ed.), (Berlin, Heidelberg), pp. 363–366, Springer Berlin Heidelberg, 1995.

[14] A. S. Lunardi, J. S. Lucena, I. R. Casella, C. E. Capovilla, and A. J. Sguarezi Filho, "Wireless communication applied in a grid tie converter control for renewable sources," 2017 6th International Conference on Renewable Energy Research and Applications, ICRERA 2017, vol. 2017-January, no. 2, pp. 552–555, 2017.

[15] National Institute of Standards and Technology, "FIPS PUB 180-2 (with Change Notice 1)," vol. 2, pp. 84, 2008.

[16] A. Menezes, P. v. Oorschot, and S. Vanstone, "Handbook of applied cryptography," pp. 755–780, 2001.

[17] V. Rijmen and E. Oswald, "Update on SHA-1," in Topics in Cryptology – CT-RSA 2005, pp. 58–71, Berlin, Heidelberg, 2005.

[18] NIST, "SHA-3 Standard: Permutation-Based Hash and Extendable Output Functions," Draft FIPS PUB 202, no. August, 2014.

[19] P. Gallagher, "Secure Hash Standard (SHA1/2) FIPS PUB 180-4," Processing, vol. FIPS PUB 1, no. October, 2012.

[20] R. P. McEvoy, F. M. Crowe, C. C. Murphy, and W. P. Marnane, "Optimisation of the SHA-2 family of hash functions on FPGAs," Proceedings - IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures 2006, vol. 2006, pp. 317–322, 2006.

[21] S. S. Omran and L. F. Jumma, "Design of multithreading SHA-1 & SHA-2 MIPS processor using FPGA," 2017 8th International Conference on Information Technology (ICIT), pp. 632-637, Amman, 2017.

[22] M. McLoone and J. V. McCanny, "Efficient single-chip implementation of sha-384 and sha-512," in 2002 IEEE International Conference on Field-Programmable Technology, 2002. (FPT). Proceedings., pp. 311–314, December 2002.

[23] F. Kahri, H. Mestiri, B. Bouallegue and M. Machhout, "An efficient fault detection scheme for the secure hash algorithm SHA-512," 2017 International Conference on Green Energy Conversion Systems (GECS), Hammamet, 2017, pp. 1-5. doi: 10.1109/GECS.2017.8066141.

[24] M. Khalil, M. Nazrin, and Y. W. Hau, "Implementation of SHA-2 hash function for a digital signature System-on-Chip in FPGA," 2008 International Conference on Electronic Design, ICED 2008, pp. 3–8, 2008.

[25] G. S. Athanasiou, H. E. Michail, G. Theodoridis, and C. E. Goutis, "Optimising the SHA-512 cryptographic hash function on FPGAs," IET Computers and Digital Techniques, vol. 8, no. 2, pp. 70–82, 2014.

[26] M. D. Rote, N. Vijendran, and D. Selvakumar, "High performance SHA-2 core using the Round Pipelined Technique,"2015 IEEE International Conference on Electronics, Computing and Communication Technologies, CONECCT 2015, pp. 1–6, 2016.

[27] S. H. Lee and K. W. Shin, "An efficient implementation of SHA processor including three hash algorithms (SHA-512, SHA-512/224, SHA-512/256)," International Conference on Electronics, Information and Communication, ICEIC 2018, vol. 2018-January, pp. 1–4, 2018.