A Dual Dynamic key chaotic Encryption System for Industrial Cyber-Physical Systems

Guangqiang Qiu\(^1\), Chengqun Wang\(^{1,2a}\), Shuyun Luo\(^1\), and Weiqiang Xu\(^1\)

Abstract

Industrial Cyber-Physical Systems (ICPS) have higher requirements on the data encryption rate and security performance; meanwhile, the devices in ICPS only support the communication protocols in the lower layers, such as the physical layer and data link layer. However, the existing encryption systems are mostly developed on the application layer and not suitable for the ICPS. Moreover, the other encryption systems, equipped the security function on the link layer, update the keys periodically, and can not satisfy the high-security requirements of ICPS. Therefore, taking the high parallel property of FPGA (Field Programmable Gate Array) into consideration, this letter proposes a Dual Dynamic key chaotic Encryption System (\(D^2\)ES) for ICPS, which provides the dynamics on the key generation and key update cycle on the data link layer. The system mainly includes an encryption module and a dual dynamic key module to realize the high-security performance. Finally, the proposed \(D^2\)ES is implemented on FPGA platform, and extensive experiments have been conducted to evaluate its security performance. The experimental results show our \(D^2\)ES provides more efficient encryption compared with the popular encryption algorithm DES (Data Encryption Standard) and can be applied for ICPS.

key words: ICPS; Data link layer; Chaotic; Dual dynamic key; FPGA

Classification: Integrated circuits (logic)

1. Introduction

Cyber-Physical Systems (CPSs) are the integration of computation, networking and physical processes and widely applied in the industrial domain. CPS in industrial infrastructures requires a higher security performance, such as the quicker encryption rate and longer key length, especially for the Operational Technology (OT) network \([1, 2, 3, 4]\). In the OT network, the diversity of supplier architectures and safety standards makes most industrial components communicate on the uncertain channels. Therefore, the OT network faces significant challenges to maintain the confidentiality and integrity of information during data transmission and storage\([5]\).

The existing encryption methods are generally divided into application-layer encryption \([6]\) and data-link-layer encryption \([7]\). On the one hand, many researchers began to study application-layer encryption. \([8]\) proposed a new cloud security technology with hybrid encryption, which combines MD5 (message-digest algorithm 5) and Blowfish encryption schemes to improve its security performance. Ratnadewi applied the Data Encryption Standard (DES) and Triple DES (TDES) cryptographic methods to protect the smart card stores data in a communication system based Near Field Communication (NFC) \([9]\). Nasution used an arithmetic coding algorithm TDES and improved the Modified least significant bit (MLSB) to achieve the image steganography of sound files \([10]\). However, most communication protocols in OT network focus on field buses, such as CAN (Controller Area Network), Modbus and EtherCAT (Ethernet Control Automation Technology), which all work at the data link layer. Hence, the above encryption strategies at the application layer are not suitable for ICPS.

On the other hand, some researchers concentrate on the encryption algorithms on the data link layer. Venkatasubramani proposed an improved quadruple Itoh-Tsujii algorithm and implemented it on the FPGA platform \([11]\). In \([12]\), a hardware design based on the conventional DES algorithm is proposed to update the key management periodically. Although dynamic keys can provide higher security performance, a corresponding attack method against 2-key TDES was proposed \([13]\), which only requires many plaintext/ciphertext pairs (even if different keys encrypt them) to reduce the key complexity and then crack a key by brute force search. Therefore, the strategy of changing the key periodically will not mitigate the attack’s success probability as expected but only reduce brute force attacks’ benefits.

Most of the communication protocol stacks for ICPS are implemented on FPGA with its useful properties of programmability, low design cost, stability, and scalability \([14, 15, 16, 17, 18, 19, 20]\). Taking the high parallel performance of FPGA into account, this letter proposes a dual dynamic key chaotic encryption system, which includes an encryption module and a dual dynamic key module. The encryption module exploits 3-key TDES to encrypt data. TDES contains two types: 2-key TDES and 3-key TDES. For simplicity, the following TDES mentioned in this letter

\(^{1}\)Zhejiang Engineering Laboratory of Industrial Internet Communication Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China

\(^{2}\)State Key Laboratory of Synthesitical Automation for Process Industries, Northeastern University, Shenyang 110004, China

a) cqwang@zstu.edu.cn

DOI: 10.1587/elex.17.20200389
Accepted November 27, 2020
Received November 19, 2020
Publicized December 07, 2020

Copyright © 2020 The Institute of Electronics, Information and Communication Engineers
refers to 3-key TDES. The dual dynamic key module adopts the MD5 to update both the key and the cryptoperiod aperiodically, making the system in a chaotic state [21, 22, 23]. Our proposed $D^2ES$ was implemented on the FPGA platform, and the experimental results demonstrated it meets the ICPS requirements on the high-speed data transmission and high-security.

The main contributions of this letter are concluded as follows.

- Taking the high parallel property of FPGA, this letter proposed a chaotic encryption system with dual dynamic keys at the data link layer, which exploits MD5 for the aperiodic key replacement to solve the shortcomings of short key length and the problem of easy cracking for fixed key in fast encryption.

- Considering FPGA’s high parallel performance, we implement our proposed $D^2ES$ and the traditional DES on FPGA. The experimental results showed that $D^2ES$ could guarantee the security requirements for ICPS without compromising the encryption rate.

The rest of the letter is organized as follows. The system model in detail is described in Section 2. The dual dynamic key module is explained in detail in section 3. The detailed algorithm flow is introduced in Section 4. The dual dynamic key encryption system Result and analysis are given in Section 5. We conclude this letter in Section 6.

2. System model

Our $D^2ES$ proposed system is based on the TDES encryption method and MD5 algorithm for dynamic keys generation. TDES can convert 64-bits plaintext into 64-bits ciphertext combined with a key length of 192 bits in each time, which has good properties, such as easy hardware implementation and high-security [24, 25, 26, 27, 1]. The MD5 algorithm can convert the plaintext into a 128-bits hash value [28, 29, 30] without reversibility, which has the strength of low collision probability.

This encryption system is divided into two modules, the TDES encryption module and the dual dynamic key module, as shown in Fig. 1. On the one hand, the TDES encryption module is composed of 7 submodules. The key submodule is responsible for generating subkeys. Next, the fifo_in submodule is used to store plaintext. Following, the padding submodule is adopted for enlarging the length of the plaintext to 64 bits. To increase the data throughput to three times, this module exploits three DES submodules as a pipelined working mode to encrypt the plaintext sequentially. Finally, the fifo_out submodule stores the output of this module, the corresponding ciphertext.

On the other hand, the dual dynamic key module contains three submodules. The control submodule is responsible for controlling the cryptoperiod and the operation of data reading. The MD5_KEY submodule focuses on updating the key, while the MD5_cycle submodule aims to update the cryptoperiod, $\mathcal{T}$.

3. Double dynamic key module

The highlight of the dual dynamic key module is a cryptoperiod update and the control state machine. The traditional encryption methods update the key periodically, which can not satisfy the security requirement for ICPS. Hence, $D^2ES$ designs the strategy of dual dynamic keys where the cryptoperiod is updated when it is due, not only the key is updated. In order to know when the cryptoperiod is due, the control state machine is exploited to detect the cryptoperiod as well as control the data reading for data integrity.

3.1 Cryptoperiod update

In the general symmetric encryption model, the sender uses the key to encrypt the plaintext and then passes the ciphertext to the receiver through the channel. After receiving the ciphertext, the receiver decrypts it with the same key to recover the plaintext. It can prevent intruders from stealing the user’s useful information directly from the channel.

As shown in Fig. 2, in addition to the plaintext and the key, this letter proposes the dynamic key module. The same $T_0$ must be given to the sender and receiver. In this system, the $\mathcal{T}$ is calculated by the number of encryptions, rather than by time. The reason is that the key will be updated even if there is no data transmission when $\mathcal{T}$ depends on time, which results in the redundant update for keys. When the time of encryption on data blocks with 64 bits reaches $\mathcal{T}$, the key will be updated. Moreover, it is possible to interrupt the encryption process if $\mathcal{T}$ is calculated by time, which brings in the key synchronization out of order. If $\mathcal{T}$ is static, the key is updated periodically. It is easy for an attacker to obtain a certain key through brute force cracking and guess the static value of $\mathcal{T}$. (To break the keys one by one.) Therefore, $\mathcal{T}$ is designed to update dynamically in our system, which makes it possible for attackers to guess all the values of $\mathcal{T}$.

Discussion: In our symmetry encryption, the same key is
needed for the encryption and the decryption. However, the key and the cryptoperiod in $D^2ES$ is dynamically updated, which puts a strict constraint on the key synchronization. Thus, our system has the threat of a low fault tolerance rate. According to the above limitation, we will introduce the communication protocol stack into $D^2ES$, where the master station updates the key and distributes it to each slave station, such that there is no extra constraint on the key synchronization.

3.2 The control state machine

Double dynamic key module needs to realize the functions of detecting cryptoperiod as well as controlling data reading. Since cryptoperiod is dynamically updated, it is necessary to judge whether the cryptoperiod is due. This module uses a counter to count the number of encryptions. Each time encryption is completed, the counter will be added by one until the cryptoperiod is reached. Meanwhile, this module should determine when to read data to guarantee the input data without loss.

![Fig. 2. The dual dynamic key encryption and decryption model](image)

As shown in Fig. 3, this state machine includes five states: S0, S1, S2, S3, and S4.

- **S0**: In this state, read a data block from the data buffer ($fifo_{in}$) for TDES encryption. When receiving Done_DES1 from the DES_1 submodule, the state changes to S1.

- **S1**: In this state, $cnt_{T} = cnt_{T} + 1$ in the encrypted data counter, and then the state is changed to S2.

- **S2**: In this state, the control module compares $cnt_{T}$ with the $T$ (key validity period) set by the system. If $cnt_{T} < T$, send the Flag_rd to $fifo_{in}$ and the state changes to S0; if $cnt_{T} >= T$, send the Flag_MD5 to MD5_key and MD5_cycle, the state transitions to S3.

- **S3**: In this state, the MD5 algorithm is used for key update and $T$ threshold update. When Done_MD5 (key update completion signal) is received, the state changes to S4.

- **S4**: In this state, the system uses the new key to generate a new subkey. When Done_key (sub-key generation completion signal) is detected, the state changes to S0.

4. The Double Dynamic Key Encryption System

The working process of $D^2ES$ is illustrated as follows. First, the data blocks are cached in the buffer ($fifo_{in}$) and be read time by time for TDES encryption. Next, TDES encryption adopts the pipeline work mode to increase the data throughput to three times. Finally, the output of our system is stored as Ciphertext in the buffer ($fifo_{out}$). Moreover, the control submodule will detect whether the number of encryptions reaches $T$. If so, it will update the key and $T$. Otherwise, it will continue encryption. The detail is described in Algorithm 1.

**Algorithm 1 The Double Dynamic Key Encryption Algorithm**

**Input**: Plaintext, Initial values of $K_1$, $K_2$, $K_3$ ($k_1$, $k_2$, $k_3$), Initial cryptoperiod ($T_0$).

**Output**: Ciphertext.

1: Receive Plaintext and cache it.

2: Subkey generation: Use the key to generate the corresponding subkey stream.
   i) Initial state: Enter $k_1$, $k_2$, $k_3$, and $T_0$.
   ii) Non-initial state: Use the new $K_1$, $K_2$, $K_3$, and $T$ generated in step 8.

3: Read plaintext from the cache.

4: Padding: Plaintext with less than 64 bits is padded with zero(s) up to 64 bits in its lower bits.

5: Data encryption.
   a) DES1: Use the subkey stream generated by $K_1$ to perform DES encryption on the output of step 4. Let $cnt_{T} = cnt_{T} + 1$.
   b) DES2: Using the subkey stream generated by $K_2$ to perform DES encryption on the output of step 5.a.
   c) DES3: Use the subkey stream generated by $K_3$ to perform DES encryption on the output of step 5.b.

6: Cache Ciphertext: Cache the output of step 5.c.

7: Cryptoperiod detection.
   i) If $cnt_{T} < T$, go to step 3.
   ii) If $cnt_{T} >= T$, go to step 8.

8: Key and $T$ update: Update the $K_1$, $K_2$, $K_3$ and $T$ by MD5 algorithm. Go to step 2.
5. Performance evaluation

To evaluate the performance of the proposed $D^2$ES, we implement the system on the FPGA platform. To compare the encryption rate, we also implement the popular encryption algorithm, DES on the FPGA platform. The extensive experimental results show that $D^2$ES has good properties of efficiency and safety by the metrics of the encryption rate and the frequency distribution.

5.1 Experiment Setting

The encryption module and the dual dynamic key module are both tested on the Quartus II 13.0 design platform. Finally, $D^2$ES was implemented on the FPGA chip of CYCLONE series, with the name of ALTERA EP4CE115F29C7N, with the model of Ethernet chip 88E1111, as shown in Fig. 4. In the whole design, a total of 43705 logic units are used.

As shown in Fig. 5, the source, Computer 1 generates the plaintext as the Output 1 and sends it to the FPGA 1 for encryption. After that, the FPGA 1 sends the ciphertext as the Output 2 to the FPGA 2 for decryption. At last, the decrypted data of the Output 3 are delivered to the destination, Computer 2. Note that, UDP communicates all the above data in this bus topology. In order to demonstrate the processes of data encryption and decryption, the output of Computer 1 (Output 1), the output of FPGA 1 (Output 2), the output of FPGA 2 (Output 3) were captured by Wireshark, shown in Fig. 5.

As shown in Fig. 6, the Output 2 shows that FPGA 1 makes the plaintext (Output 1) become a piece of meaningless data. At the same time, its original information is the text of three consecutive "computer"s. It is easy to observe that the ciphertext of Output 2 contains the same data for the first two "computer”, and the different data for the third "computer". The reason is that the $T_0$ value is set to 2 in our experiment. That means the key will be changed after the two plaintext blocks (64 bits) are encrypted. Hence, the ciphertext of the third "computer" is changed, which confirms the dynamic key function. In addition, Output 3 is consistent with the

Output 1. The whole process confirms that our $D^2$ES is feasible on the FPGA platform.

5.2 The security evaluation

The proposed $D^2$ES exploits a dual chaotic mechanism to guarantee the system in an uncertain state all the time. The updated key causes the first chaotic. The second chaotic is introduced by dynamic cryptoperiod. Even if an attacker cracks the key within a certain period, the original rule of cracking key will be failed since the dynamic cryptoperiod. The attacker needs to find another rule of cracking key, which is time-consuming and impossible to be obtained by the updated time of the cryptoperiod.

For simple description, the bit length of $T$ is defined as $n$. To evaluate the security performance with different $n$, we record the distribution frequency of each letter in the plaintext and the ciphertext in Fig. 7. In the $D^2$ES, we design that one letter is encrypted into two hexadecimal numbers with 8 bits. Hence, there are $2^8 = 256$ cases, divided into 16 groups, each of which has 16 cases.

In general, we use the frequency distribution of each letter to evaluate the hardness of key cracked. Because the more evenly of the frequency distribution, the harder to crack the ciphertext. Fig. 7(a) shows the frequency distribution of plaintext. (how many plaintext packets are recorded.) Fig. 7(b), Fig. 7(c), and Fig. 7(d) are the frequency distributions of the ciphertext with $n = 4$, 6, and 8, respectively. From Fig. 7, it is observed that the frequency distribution are relatively uniform in ciphertext compared with the one in plaintext.

In order to evaluate the uniformity with different $n$, the standard deviation of each group is calculated as $\sigma_1 = 6.67 \times 10^{-3}$, $\sigma_2 = 7.05 \times 10^{-3}$, and $\sigma_3 = 9.43 \times 10^{-3}$. Thus, larger $n$ makes a high deviation of frequency distribution, but with little difference. The reason is that larger $n$ makes the generated key with a more extended validity period. Therefore, more repeated words in the plaintext will
also have an apparent high frequency in the ciphertext. While
$n$ is short, the key is updated frequently, which makes the
property of the plaintext without effect on the ciphertext.
The encryption of $D^2ES$ has a high-security performance
by the highlight of a dual chaotic mechanism.

5.3 The efficiency evaluation
The cryptoperiod $T$ has an excellent impact on the encryp-
tion rate. Specifically, the small value of cryptoperiod brings
the high frequency of key update, while the big one results
in the low performance of system security.
It is crucial to determine an appropriate $n$ for the tradeoff
between the encryption rate and system security. Simulated
on ModelSim with 400MHZ clock, the encryption rate of the
proposed $D^2ES$ are recorded under the settings of $n$
equals to 4, 6, and 8, compared with the traditional DES
algorithm. The experimental results are shown in Table I.
$D^2ES$ can reach an encryption time of nanoseconds to match
the transmission requirements for ICPS.

| Table 1. Encryption Performance Experimentation Results |
|---|
| Data Length (byte) | DES (ns) | Double dynamic key (ns) |
| n=4 | n=6 | n=8 |
|---|---|---|
| 8 | 197.5 | 540 | 540 | 540 |
| 16 | 395 | 970 | 970 | 745 |
| 32 | 770 | 1360 | 1360 | 1162.5 |
| 64 | 1542.5 | 2162.5 | 2162.5 | 1967.5 |
| 128 | 3087.5 | 3992.5 | 3767.5 | 3567.5 |
| 256 | 6180 | 7655 | 6980 | 6772.5 |
| 512 | 12385 | 14775 | 13650 | 13225 |
| 1024 | 24767.5 | 29437.5 | 26962.5 | 26087.5 |
| 2048 | 49530 | 61422.5 | 53360 | 51807.5 |
| 4096 | 99212.5 | 119542.5 | 106562.5 | 103652.5 |

For a more vivid illustration, the encryption time ratio of
the proposed $D^2ES$ and the traditional DES algorithm is defined
as $\lambda$. The baseline of Fig. 8 is the encryption efficiency
of the traditional DES algorithm, listed in the second column
of Table I. The ratios are symboled as $\lambda_1$ with $n = 4$, $\lambda_2$
with $n = 6$ and $\lambda_3$ with $n = 8$, respectively. As shown in
Fig. 8 when the length of the plaintext is 8 bytes, $\lambda_1$, $\lambda_2$
and $\lambda_3$ are the same as 2.734. The reason is that the short
length of the plaintext fails to reap the benefit of the pipeline
mechanism. As the plaintext’s length increases, three DES
models can conduct the encryption in the parallel mode un-
der the pipeline mechanism, which improves the encryption
efficiency. When the length of the plaintext increases to 4096
bytes, $\lambda_1$, $\lambda_2$, and $\lambda_3$ are approximate to 1, i.e., 1.205, 1.074,
and 1.045, respectively. This is because the high update
frequency brings time cost when $n = 4$, resulting in a signif-
ican bottleneck in the encryption time. When $n$ increases
6 or 8, the decreased frequency of the key update makes little
impact on the encryption time. From Fig. 8, it is easy to
observe that the encryption time is almost the same as one
of the DES algorithms in the case of the extensive length
of plaintext and $n = 6$ or $8$. Taking the above analysis
into consideration, the proposed $D^2ES$ has a high-security
performance with a good encryption time.

6. Conclusion

In this letter, we have proposed a dual dynamic key chaotic
encryption system ($D^2ES$) for ICPS with the dynamic key
and cryptoperiod, which makes an attacker challenging to
rack the key within the cryptoperiod. It can encrypt the data
at the data link layer. $D^2ES$ is implemented on the FPGA
platform and conducted extensive experiments. The experi-
mental results show our $D^2ES$ provides higher security with-
out compromising the encryption efficiency, compared with
the popular encryption algorithm DES. Therefore, the pro-
posed $D^2ES$ can be applied for ICPS.

Acknowledgments

This work is supported in part by the National Natural
Science Foundations of China (No. U1709219 and No.
61701444), the Key Research and Development Program
of Zhejiang Province 2018C01093, and Liaoning Provincial
Natural Science Foundation of China (2020-KF-21-09).
References

[1] IMAI, et al.: “Development of Cryptology in the Nineties,” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences 84.1 (2001): 61-67.

[2] A. Karati, et al.: “Provably Secure and Generalized Signcryption With Public Verifiability for Secure Data Transmission Between Resource-Constrained IoT Devices,” IEEE Internet of Things Journal, (2019) (DOI: 10.1109/JIOT.2019.2939204).

[3] K. Sowjanya, et al.: “An Efficient Elliptic Curve Cryptography-Based Without Pairing KPABE for Internet of Things,” IEEE Systems Journal, (2019) (DOI: 10.1109/JSYST.2019.2944240).

[4] S. Atiewi, et al.: “Scalable and Secure Big Data IoT System Based on Multifactor Authentication and Lightweight Cryptography,” IEEE Access, (2020) (DOI: 10.1109/ACCESS.2020.3002815).

[5] Barker, et al.: “Recommandation for the triple data encryption algorithm (TDEA) block cipher,” No. NIST Special Publication (SP) 800-67 Rev. 2 (Draft). National Institute of Standards and Technology (2017).

[6] Fu, et al.: “A New Efficient Algorithm for Secure Outsourcing of Modular Exponentiations,” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences 103.1 (2020): 221-224.

[7] A. PéRez-Resa, et al.: “Self-Synchronized Encryption for Physical Layer in Gigabit Ethernet Optical Links,” IEEE Access(2020) 2993550 (DOI: 10.1109/ACCESS.2020.2993550).

[8] A. Chauhan and J. Gupta: “A novel technique of cloud security based on hybrid encryption by Blowfish and MD5,” 2017 4th International Conference on Signal Processing, Computing and Control (ISPCC), Solan (2017) 8269702 (DOI: 10.1109/ISPCC.2017.8269702).

[9] Ramadewi, R. P: “Implementation cryptography data encryption standard (DES) and triple data encryption standard (3DES) method in communication system based near field communication (NFC),” J. Phys. Conf. Ser. Vol. 954. No. 1. (2018).

[10] Nasution, et al.: “Image steganography in securing sound file using arithmetic coding algorithm, triple data encryption standard (3DES) and modified least significant bit (MLSB),” International Conference on Mechanical, Electronics, Computer, and Industrial Technology (MECnIT) (2018).

[11] V. R. Venkatasubramani, et al.: “An improved quad Itoh-Tsujii algorithm for FPGAs,” IEICE Electronics Express (2013) (DOI: 10.1587/elex.201303612).

[12] J. Yao and H. Kang: “FPGA implementation of dynamic key management for DES encryption algorithm,” Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology, Harbin (2011) (DOI: 10.1109/EMEIT.2011.6024111).

[13] C. J. Mitchell: “On the Security of 2-Key Triple DES,” IEEE Transactions on Information Theory (2016) 2611003 (DOI: 10.1109/TIT.2016.2611003).

[14] Toshihiro KATASHITA, et al.: “Development of an Evaluation Platform and Performance Experimentation of Flex Power FPGA Device,” IEICE TRANSACTIONS ON Information and Systems, (2018) (DOI: 10.1587/transinf.2017RCP0003).

[15] J. Zhang, et al.: “A PUF-FSM Binding Scheme for FPGA IP Protection and Pay-Per-Device Licensing,” IEEE Transactions on Information Forensics and Security, (2015) (DOI: 10.1109/TIFS.2015.2400413).

[16] B. Koziel, et al.: “Post-Quantum Cryptography on FPGA Based on Isogenies on Elliptic Curves,” IEEE Transactions on Circuits and Systems, (2017) (DOI: 10.1109/TCSC.2016.2611561).

[17] S. P. Guruprasad and B. Chandrasekar: “An evaluation framework for security algorithms performance realization on FPGA,” 2018 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC), (2018) (DOI: 10.1109/ICCTAC.2018.8370396).

[18] G. Gallin and A. Tisserand: “Generation of Finely-Pipelined Gfi PP) Multipliers for Flexible Curve Based Cryptography on FPGAs,” IEEE Transactions on Computers, (2019) (DOI: 10.1109/TC.2019.2920352).

[19] M. M. Islam, et al.: “FPGA Implementation of High-Speed Area-Efficient Processor for Elliptic Curve Point Multiplication Over Prime Field,” IEEE Access, (2019) (DOI: 10.1109/ACCESS.2019.2958491).

[20] Fang Zhou, et al.: “A new method for resisting collision attack based on parallel random delay S-box,” IEEE Electronics Express, (2019) (DOI: 10.1587/eleex.16.20190192).

[21] Matthews, Robert: “On the derivation of a "chaotic" encryption algorithm,” Cryptologia 13.1 (1989): 29-42.

[22] Z. Hua, et al.: “Sine Chaoticity Model for Enhancing Chaos and Its Hardware Implementation,” IEEE Transactions on Industrial Electronics, (2019) (DOI: 10.1109/TIE.2018.2833049).

[23] M. Sharafi, et al.: “A Low Power Cryptography Solution Based on Chaos Theory in Wireless Sensor Nodes,” IEEE Access, (2019) (DOI: 10.1109/ACCESS.2018.2886384).

[24] Y. Wei, et al.: “New second-order threshold implementation of AES,” IET Information Security (2019) 8724111 (DOI: 10.1049/iet-ifs.2019.5244).

[25] H. Tang, et al.: “A Network Coding and DES Based Dynamic Encryption Scheme for Moving Target Defense,” IEEE Access, (2018) (DOI: 10.1109/ACCESS.2018.2832854).

[26] Patil, Priyadarshini: “A comprehensive evaluation of cryptographic algorithms: DES, 3DES, AES, RSA and Blowfish,” Procedia Computer Science 78.1 (2016): 617-624.

[27] Mankotia, et al.: “A critical analysis of some symmetric key block cipher algorithms,” International Journal of Computer Science and Information Technologies 6.1 (2015): 495-499.

[28] K. N. Sreehari and R. Bhaktavatchalu: “Implementation of hybrid cryptosystem using DES and MD5,” 2018 3rd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India (2018) 8724111 (DOI: 10.1109/CESYS.2018.8724111).

[29] J. Jiménez, et al.: “Power and Area Reduction of MD5 based on Cryptoprocessor Using novel approach of Internal Counters on the Finite State Machine,” 2019 IEEE Fourth Ecuador Technical Chapters Meeting (ETCM), Guayaquil, Ecuador (2019) 9014878 (DOI: 10.1109/ETCM48019.2019.9014878).

[30] K. Quist-Apheti and M. C. Xenya: “Node to Node Secure Data Communication for IoT Devices Using Diffie-Hellman, AES, and MD5 Cryptographic Schemes,” 2019 International Conference on Cyber Security and Internet of Things (ICSIoT) (2019) 00022 (DOI: 10.1109/ICSIoT.2019.20190192).

[31] M. Gillela, et al.: “Parallelization of Brute-Force Attack on MD5 Hash Algorithm on FPGA,” 2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID), (2019) (DOI: 10.1109/VLSID.2019.200034.2886384).