IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 3, 2020, accepted February 22, 2020, date of publication March 5, 2020, date of current version March 18, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.2978665*

# Architectural Optimization of Parallel Authenticated Encryption Algorithm for Satellite Application

**SYED JAHANZEB HUSSAIN PIRZADA, ABID MURTAZA, TONGGE XU, AND LIU JIANWEI**
School of Cyber Science and Technology, Beihang University, Beijing 100083, China

Corresponding author: Syed Jahanzeb Hussain Pirzada (jahanzebp@hotmail.com)

This work was supported by the Industrial Internet Innovation and Development Project of the Ministry of Industry and Information Technology (Grant no. IIIDP-9.1-2018 of MIIT).

**ABSTRACT** High-speed data communication is becoming essential for many applications, including satellite communication. The security algorithms associated with the communication of information are also required to have high-speed for coping up with the communication speed. Moreover, the Authenticated Encryption (AE) algorithms provide high-speed communication and security services include data encryption, authentication, and integrity. The AE algorithms are available with serial and parallel architectures; among them, the Galois Counter Mode (GCM) algorithm has a parallel architecture. The Synthetic Initialization Vector (SIV) mode in the AES-GCM-SIV algorithm provides the nonce misuse protection using the GCM algorithm. Besides, reduced data throughput is provided using the AES-GCM-SIV algorithm as compared to the AES-GCM algorithm. This work introduced a parallel algorithm with re-keying and randomization of the initialization vector for high data throughput, nonce misuse protection, and side-channel attack protection. The implementation of the proposed algorithm is performed on Field Programmable Gate Array (FPGA) and it's compared with the FPGA implementations of AES-GCM, AES-GCM-SIV, and recently introduced algorithms. The optimization of the proposed algorithm and security analysis is presented for space application using different optimizations and a combination of optimizations.

**INDEX TERMS** Authenticated encryption, FPGA, nonce misuse attack, parallel architecture, satellite communication, side-channel attack.

## I. INTRODUCTION

The high-speed communication systems are required because of high-speed computational requirements for providing extended facilities for digital system consumers. The application of high-speed equipment varies from a cell phone communication system, including the usage of 5G communication technology to satellite communication with extended services for many applications, including the imaging services, even from a GEO satellite [1]. The amount of data required for different applications using a satellite has increased because the satellite data is used in almost all kinds of applications, including disaster management [2], air traffic control [3], and many other applications. The main reason for the increase in the utilization of communication services offered by satellites is due to its data throughput, onboard data storage, and the number of communication links. The challenges for communication systems are equally valid for the data security algorithms associated with it for the provision of secure communication. Data security algorithms provide three primary services, such as data confidentiality, authenticity, and integrity. The utilization of these services is associated with the type of algorithm used in an application; for instance, data encryption algorithms provide the confidentiality service. Data authentication algorithms offer authenticity and integrity services. The Authenticated Encryption (AE) algorithms provide confidentiality, authenticity, and integrity services. The Consultative Committee for Space Data Systems (CCSDS) and European Space Agency (ESA) recommended the utilization of these services as per mission requirements [4]. The increase in the usage of data from satellites has compelled the use of a data security algorithm that can ensure high-level security and reliability of data. The AE algorithms provide all the primary

The associate editor coordinating the review of this manuscript and approving it for publication was Yassine Maleh.


This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

three services required for guaranteeing the high-security and reliability of data. Besides, the former challenge for the provision of high-speed communication systems remains a difficulty, owing to the selection of high-speed security algorithms. Previously, serial architecture AE algorithms such as Advanced Encryption Standard in Counter mode (AES-CTR) with Cipher-block Chaining Message authentication code (AES-CCM) algorithm [5] was utilized for the provision of data security for many applications. The AES-CCM algorithm uses the AES-CTR algorithm with the Message Authentication Code (MAC) based authentication algorithm. Although the CCM algorithm has limited data throughput and is not a recommended option for high data throughput applications such as satellite communication. Also, the parallel architecture AE algorithms, such as Galois Counter Mode (GCM) algorithm [6], [7], are increasingly utilized in various applications. The CCSDS recommends the GCM algorithm for satellite applications [8]. The parallel architecture of the GCM algorithm provides the advantage of high data throughput. The implementation and comprehending the design becomes easier as it utilizes the state-of-the-art AES-CTR algorithm with Hash-based authentication. The famous AE algorithms such as AES-CCM and AES-GCM use the AES-CTR algorithm.

In the AES-CTR algorithm, the Initialization Vector (IV) consists of a nonce, constant vector, and counter vector. Besides, the repetition of IV can lead to compromising the security of the algorithm; therefore, the IV is incremented by one for generation of every ciphertext. Although the nonce vector remains the same and the increment is only in the counter vector. Recently, the security of the GCM algorithm has become questionable with attacks based on the fact that the IV has a constant nonce [9]. If the constant nonce used for two messages, the plaintext could be leaked. The use of exclusive-OR operation of plaintext with keystream to generate ciphertext with constant nonce will create security concerns of nonce misuse. Previously, the Cipher-based Message Authentication Code (CMAC) algorithm is used in the AE algorithm for the generation of Synthetic Initialization Vector (SIV) [10]. Where the CMAC algorithm was used for authentication of Additional Authentication Data (AAD) and the AES-CTR algorithm was used for encryption and tag generation. In addition, the optimization of the AES-GCM algorithm is proposed in the AES-GCM-SIV algorithm for nonce misuse resilient implementation [11]. The implementation provides the nonce misuse protection on the cost of reduction of data throughput. Recently, a researcher provided a comparison between the AES-GCM and AES-GCM-SIV algorithms implementation on FPGA [12].

In 2012, a competition known as Competition for Authenticated Encryption: Security, Applicability, and Reliability (CAESAR) were arranged for a selection of AE algorithms to face future challenges. The CAESAR competition involves algorithms for catering the nonce misuse problem and the selection of new algorithms for modern-day high-speed communication [13]. After the evaluation of the algorithms submitted in the competition, a total of six algorithms are recommended for AE of data for different applications. The proposed algorithms in CAESAR competitions are used for various applications as well as comparative studies have been carried out for different applications, including Internet of Things (IoT) [14]. Besides, different researchers proposed the weakness present in the algorithms proposed for CAESAR [15]. Some of the proposed schemes in CAESAR have better computation speed as compared to the AES-GCM algorithm, but still, some schemes lag in different aspects such as they are not nonce misuse resilient. Some of the finalists of CAESAR competition are entirely, and some are partially nonce misuse resilient. The research on the implementation of different proposed algorithms in CAESAR competition has revealed that there is a tradeoff between efficiency and resource utilization of different algorithms for achieving nonce misuse protection. For instance, the PRIMATE APE algorithm [16] has a smaller area, but it is lower in data throughput as compared to other nonce misuse resilient algorithms.

Besides the nonce misuse attacks, passive attacks such as the side-channel attacks are becoming a point of concern for researchers to enable flawless security for AE algorithms. The side-channel attacks are based on monitoring of physical parameters (such as current measurement, electromagnetic emission, etc.). The side-channel attacks are more prominent on AE algorithms due to the utilization of a single key in a security algorithm. Many block ciphers proposed in CAESAR competition utilize a single key. The use of a single key by many proposed algorithms in CAESAR has made them susceptible to side-channel attacks [17] and Differential Power Analysis (DPA) attacks [18]. These attacks are related to physical contact with the hardware for the analysis of emitted power dissipation. This kind of attack monitors the power dissipation or the electromagnetic field utilization of the device for secret key recovery. Recently, a researcher has shown that many algorithms such as AES-GCM, Ascon, ACRON, CLOC, JAMBU, SILC, and Ketje Jr. are affected by DPA while implementing these algorithms on FPGA [19]. Therefore, countermeasures are to be adopted to cater to the effects of these attacks. The methods for countermeasure involves the masking the cryptographic algorithm [20] and re-keying algorithm [21]. The masking process consists of the hiding of the algorithm, but it adds up overhead to the algorithm [22]. The re-keying process involves generating a fresh session key for achieving inherent protection against these attacks. The Perfect Forward Secrecy (PFS) uses the approach of re-keying, where every message uses a new key. The re-keying and forward secrecy concepts are being used rapidly in different applications. As in case of security leakage, only one message is compromised.

On the other hand, the environment in space has adverse effects on the electronics devices as compared to the environment on earth. The space environment contains high pressure, vacuum, and radiations that create an exceptional condition of operation for electronics hardware. Therefore,

special measures are to be taken to protect the electronic devices to withstand the space environment [23]. In addition, the optimization of algorithms implemented on the device is required for errorless operation of algorithms in the space environment. Especially the security algorithms such as AE algorithms must be optimized for operation in space to ensure reliable and secure communication. The radiations in space effects memories in electronic hardware; the effects are called Single Event Effects (SEE) [24], [25]. These effects force the algorithm to malfunction. Besides, there are many SEE error mitigation techniques, but the modification in architecture for optimizing it for AE algorithms helps to reduce SEE without inducing an overhead. Researchers proposed that designing the AE algorithm by avoiding the use of memories such as Static Random Access Memory (SRAM) can help in reducing the SEE [26]. Fortunately, the design of security algorithms (such as the AES algorithm) uses less amount of memory. Especially in AES algorithm memory is used for storing the Substitution box (S-box) table. The implementation of the algorithm, such as S-box instead of using the table, can help to avoid the memory utilization. Hence, the requirement of high-speed and secure AE algorithm has particular challenges as follow.

- The algorithm must be a parallel architectural algorithm for the provision of high data throughput.
- The algorithm must be nonce misuse resistant for providing secure communication.
- The algorithm must be side-channel attack resilient such as DPA, for providing secure communication.
- The architectural optimization for space environment should be implemented for the operations of security algorithms in space.

The present algorithms do not fulfill the challenges mentioned above. In case some algorithm fulfills the high throughput requirement (such as AES-GCM); they do not provide nonce misuse resilient property. On the other hand, in case an algorithm provides nonce misuse resilient property (such as AES-GCM-SIV); it do not have high data throughput. Also, the re-keying or masking requirement to get rid of the side-channel attacks is not implemented in famously used algorithms such as the GCM algorithm. Similarly, the architectural optimizations are required for the optimization of the algorithm for many applications, including the satellite application. Therefore, in this work, a parallel architecture AE algorithm is proposed for high data throughput along with nonce misuse protection, side-channel attack resistant, and optimization for satellite application using the AES-CTR algorithm.

The rest of the paper is organized as follows; section II provides the related work completed for coping with the challenges on security algorithms. Section III gives details about the proposed algorithm architecture and design. Section IV presents the architectural optimization for satellite applications. Section V elaborates on the experimental results of the implementation of the proposed algorithm and its comparison with the famous algorithms along with architectural optimization. Section VI analyzes the details of the discussion on the security analysis of the proposed algorithm. At last, Section VII concludes the paper.

## II. RELATED WORK

The AE algorithms have been researched and developed over a while. The increasing high-speed communication requirement and growing security attacks concern the researchers for the development of new algorithms with better performance and security coverage. The AES-GCM algorithm and its enhancement for nonce misuse protection in the AES-GCM-SIV algorithm show the evolution of AE algorithms due to effectiveness of attacks on security algorithms. Besides, many researchers have provided the optimization of the AES-GCM algorithm by efficient utilizing area [27]–[29] and by increasing throughput [30], [31]. Meanwhile, a Parallel Cipher-based Message Authentication Code (PCMAC) algorithm was introduced for fulfilling the high throughput requirements of modern-day computation [32]. The PCMAC algorithms provide comparatively similar throughput as compared to the AES-GCM algorithm, although it utilized MAC-based authentication instead of Hash-based authentication. Also, a PCMAC Synthetic Initialization Vector (PCMAC-SIV) algorithm is proposed for nonce misuse resistant implementation [33]. The PCMAC-SIV algorithm provide comparable throughput as compared to the AES-GCM-SIV algorithm. The proposed AE algorithms in CEASAR competition have also proposed nonce misuse resistant algorithms, and even some algorithms provide better throughput as compared to the AES-GCM algorithm. The pipeline able On-line Encryption with the authentication Tag (POET) algorithm [34] proposed an AE algorithm using four rounds of AES algorithm. The POET algorithm is lightweight and suitable for the Internet of Things (IoT) and many other applications. The POET algorithm is implemented in parallel and has nonce misuse resilience property. The POET algorithm low resource requirements and parallel implementation with fewer clock cycles enabled it to provide high data throughput.

Recently, with the increasing demand of AE algorithms for high data throughput and nonce misuse protection, many researchers have provided a comparison between the candidates of CEASAR competition [14]. The comparative implementation is mainly performed using Field Programmable Gate Array (FPGA) hardware using its parallel implementation capability. The Deoxys algorithm implementation comparison is provided with the AES-GCM algorithm using the FPGA hardware platform [35], and many other algorithms performed similar comparisons [32]. Therefore, in this work, the FPGA hardware platform is selected for implementation of the proposed algorithm for providing high data throughput, nonce misuse resilient implementation, and side-channel attacks immune algorithm.

The nonce misuse attack is due to utilization of constant nonce in IV of block cipher algorithms such as

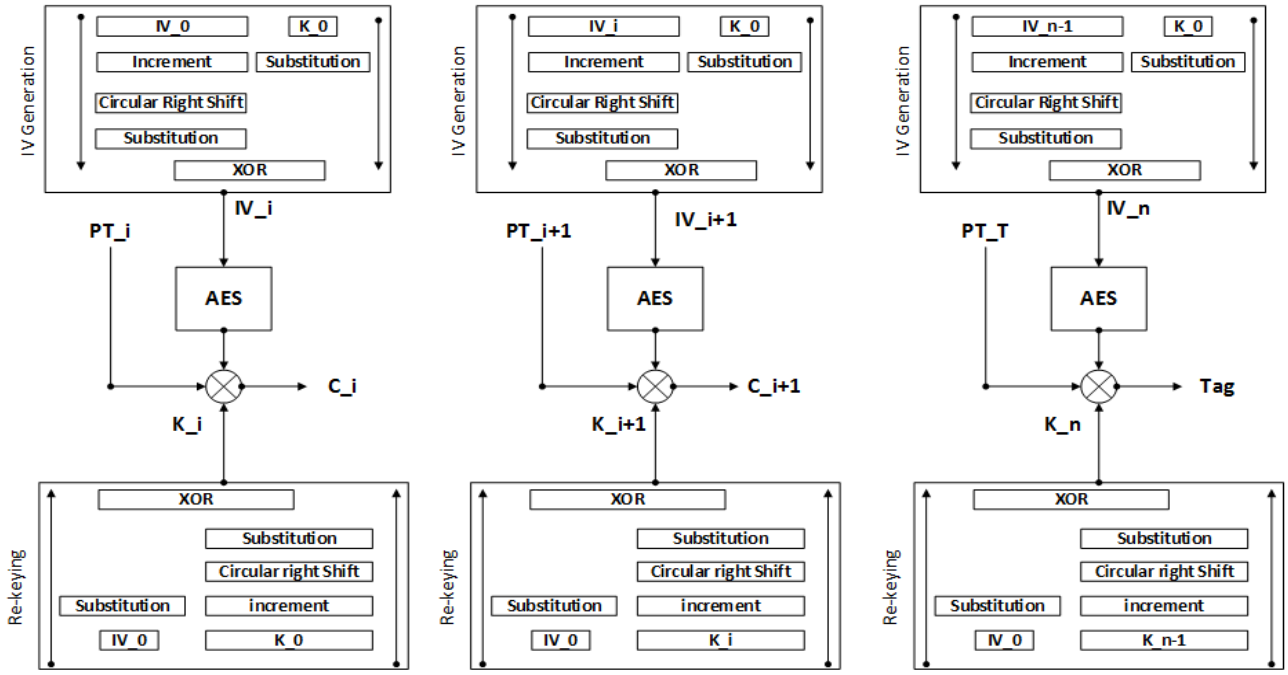


**FIGURE 1.** Block diagram of the proposed parallel nonce misuse resistant AE algorithm. The first two blocks are used for the generation of ciphertext, and the last block is used for the generation of the authentication tag.

AES-CTR algorithm. Some authors have offered their methodologies to modify the IV to generating randomness in IV. Recently, one of such methods proposed the use of GEFFE generator with incrementing the IV with one used for satellite images for the randomization of ciphertext [36]. Similarly, researchers have proposed to use the right shift and increment operation for randomization of IV [37]. The generated ciphertext was tested for the National Institute of Standards and Technology (NIST) run and frequency test for checking the randomization of generated ciphertext. Also, side-channel attacks have shown its effectiveness on different algorithms, including the AES algorithm [38]. Also, the block cipher based AE algorithms such as the AES-GCM algorithms are affected by the side-channel attacks. Recently, the DPA attack has affected many AE algorithms, including the AES-GCM algorithm [19]. The re-keying algorithm can protect AE algorithms against the DPA attacks. The re-keying algorithm involves using a different key for every session for generation of ciphertext [39]. In this work, the proposed algorithm provides the IV randomization algorithm for the nonce misuse resistant method and a re-keying algorithm for protecting the data from side-channel attacks such as the DPA attack.

Many researchers have proposed the architectural optimizations for implementing the AES-GCM algorithm on the FPGA hardware. Also, the new algorithms proposed in CEASAR competition have been implemented on FPGA using some proposed architectural optimizations [40]. The architectural optimizations for satellite applications are more or less similar to the optimizations performed on systems on the ground. Besides, there are some extreme conditions in space environment that have compelled to perform additional architectural optimization for satellites. These extreme conditions are the radiation on space due to Van Allen radiation belt and solar flares. Many researchers proposed various optimizations for implementation to reduce the effects of the radiations on the electronics hardware [26]. The utilization of this algorithm for satellite applications has compelled us to propose some additional architectural optimization for the proposed AE algorithm.

## III. THE PROPOSED AE ALGORITHM

The proposed algorithm has three parts; the IV generation algorithm, the block-cipher, and the re-keying algorithm. The IV generation algorithm randomizes the IV for providing protection against nonce misuse attack. The second part uses a block cipher algorithm, any block cipher algorithm can be utilized, but we have selected the AES-CTR algorithm. The third part is the re-keying algorithm, which protects against side-channel attacks such as the DPA attack. The re-keying algorithm provides a separate key for every session for generating a ciphertext, which makes the side-channel attacks ineffective against the proposed algorithm. The block diagram of the proposed algorithm is in Figure-1. The proposed algorithm can use key of any size such as 128-bits, 192-bits and 256-bits. Besides, in this work, the key size of 128-bits is implemented. The PT_T denotes the exclusive-OR of all the plaintexts, the exclusive-OR product of plaintexts will be utilized for the generation of authentication tag. The IV and

secret key are pre-shared amongst the users for decryption and authentication of data.

### A. THE IV GENERATION ALGORITHM

The nonce misuse protection can be performed by modifying the nonce for every plaintext, which was left constant in many famous AE algorithms such as the GCM algorithm. Also, many researchers have proposed the modification of IV for its randomization [36], [37]. The extent of its randomization was verified by using the NIST run and frequency tests. In the proposed AE algorithm, we propose a similar approach of pre-processing the IV for randomization of IV. The IV is generated using a secret IV generated randomly. Besides, the generic IV composed of constant vector, nonce, and counter vector can be used as well. Three processes are applied on IV, which are increment the IV by one, circular right shift of IV, and substituting value of IV using Substitution Box (S-Box). The S-Box used for IV generation is the same as used in the AES algorithm. The IV is firstly incremented with one for every plaintext. Then the incremented IV is circular right shifted for rotating the IV. The shifted IV is then substituted using an S-Box. At last, the key generated from re-keying algorithm will be used in the IV generation algorithm. The key is substituted using the S-Box algorithm, and then it's exclusive-OR with the substituted IV for the generation of a new IV. The new IV generated (i.e. IV1) is input to the IV generation algorithm for the generation of next ciphertext. In contrast, the initial secret key for rekeying will be input for IV generation for the generation of all ciphertexts. The flow of the IV generation algorithm is shown in Figure 2.

### B. THE BLOCK-CIPHER ALGORITHM

The encryption of plaintext is performed using a block cipher algorithms; in the proposed algorithm, AES-CTR is used for encryption of the plaintext. The AES-CTR algorithm has a unique property of using a parallel architecture implementation.

The parallel architecture is the reason for its utilization in many high data throughput AE algorithms. Therefore we have also utilized the AES-CTR algorithm for encryption of plaintext in the proposed algorithm. The proposed algorithm uses IV, secret key, and plaintext (PT) as an input and it generates the ciphertext and authentication tag (Tag) as an output. The secret key used as input to AES-CTR algorithm as well as in the IV generation and re-keying algorithm.

### C. THE RE-KEYING ALGORITHM

The side-channel attacks are very prominent attacks affecting the AE algorithms. The protection against side-channel attacks has become essential to ensure reliable and secure communication. Many techniques are proposed for the protection of data from side-channel attacks using data masking, re-keying, etc. The proposed algorithm utilizes the re-keying algorithm for protecting the algorithm against the side-channel attacks. The proposed algorithm use the initial secret key (K_0) as an input and generated another key for each session (K_i). The secret key is incremented by one, circular right-shifted, and then it's substituted by using S-Box. The initial IV (i.e. IV_0) is taken as input in the re-keying algorithm. The initial IV is substituted using S-Box and then exclusive-OR with the substituted original key. The new key generated will be the key for the next block. The algorithm for the IV generation and the re-keying algorithm is the same, except the inputs are different.

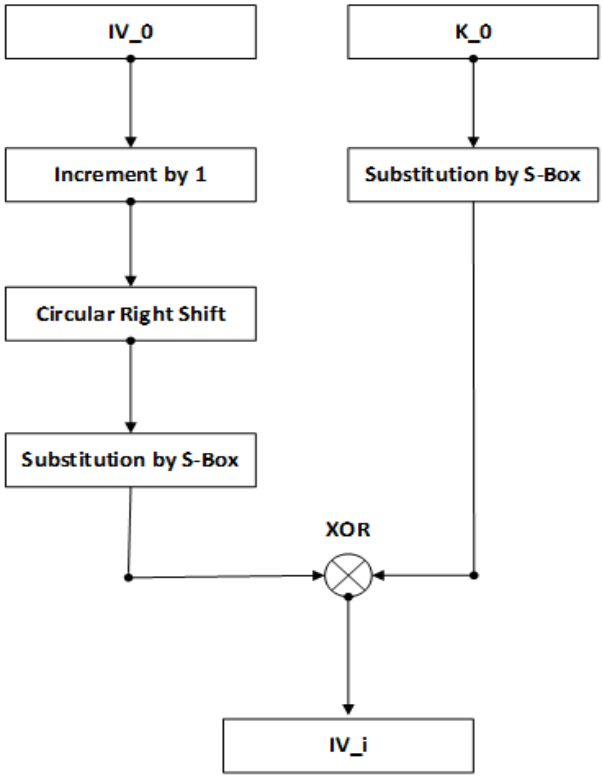


**FIGURE 2.** The flow chart of the proposed IV generation algorithm.

## IV. THE IMPLEMENTATION METHODOLOGIES AND ARCHITECTURAL OPTIMIZATION

This section presents the implementation methodologies available for the implementation of AE algorithms on FPGA. The discussion of the methods for implementation is significant in understanding the pros and cons of each technique. Also, different architectural optimizations are presented to implement the proposed AE algorithm for satellite applications.

### A. THE IMPLEMENTATION METHODOLOGIES

There are different implementation methodologies available for implementing any algorithm on FPGA, including the AE algorithms. The methods include the Hardware Description Language (HDL) core design, Intellectual Property (IP) Core Design, and High-Level Synthesis (HLS) based core design.

#### 1) THE HDL CORE DESIGN

The HDL core design involves the design of the algorithm using Verilog HDL and VHDL languages [41], [42].

The design is performed by an individual designer or research group for their usage or distribution under the open license agreement. The design has many advantages such as control over the modification of algorithm, implementable on FPGA of any vendor, modification of algorithm for utilization for research purpose, implementing architectural optimization, no licensing issues, and many more.

#### 2) THE IP-CORE DESIGN

The IP core design involves the design of the algorithm using the pre-defined IP cores provided by either the FPGA vendor or by the third party company. The IP cores include the implementation of famous algorithms along with their optimized implementation for a specific FPGA family of a particular vendor. The IP core based design help in quick prototyping and highly optimized implementation. Besides, the IP core design is for a specific vendor and cannot be used with another FPGA vendor; the IP core has limited controllability in the modification of some parameters. The IP core cannot be used for research purposes, the architectural modification cannot be applied, and many more. The disadvantages of the IP core design have impacted the utility of IP cores for many applications, including research. Recently, a comparison between the HDL core design and the IP core design is presented for their utility in different applications [43].

#### 3) THE HIGH-LEVEL SYNTHESIS BASED DESIGN

The HLS based design involves the design of algorithms using software tools such as the MATLAB software. The HLS based design is performed by programming an algorithm using a high-level language such as C language and then converting the program in C language into HDL. Many FPGA vendors, including Xilinx, Microsemi Actel, and many more, have provided their tools to import the code in MATLAB to be implemented in their FPGAs [44], [45]. The implementation through such a method can help to implement a complex algorithm quickly by using a pre-programmed algorithm from MATLAB and other software on FPGA. The algorithm provides controllability on algorithm design, the architectural optimizations can be implemented on the algorithm, and the design can be implemented in any vendors FPGA whose tool is integrated with MATLAB or provided by the vendor. Besides, the disadvantage of this methodology is implementation is not optimized as per the area utilization. The implementation of different algorithms using the HLS is presented by various researchers [46], [47].

### B. THE ARCHITECTURAL OPTIMIZATION

There are many architectural optimizations presented for optimization of area utilized, throughput enhancement, and many other goals as per the area of application. In this work, we are focused on architectural optimization for satellite applications to be utilized in space and on the ground. The implementation on the ground is quite similar to generic architectural optimizations, although the architectural optimization for satellites in space has different goals. The space environment is affected by vacuum conditions, high pressure, and especially radiation. Therefore the optimizations are required to be implemented, taking these effects in mind. The main requirements are to have high data throughput for fast communication and implementation for catering the issues in space environment.

#### 1) PARALLEL ARCHITECTURE IMPLEMENTATION

The proposed AE algorithm is parallel in architecture and using the AES-CTR algorithm, which does not have chaining or feedback mode for providing high data throughput. The parallel implementation requires fewer clock cycles for design implementation. Besides, the algorithms with chaining mode or feedback mode are usually optimized using the pipelining method. The proposed algorithm already has a parallel implementation, although it will consume more area for implementation as compared to the serial implementation. The proposed algorithm requires few clock cycles for the generation of IV and key, in addition to the clock cycles needed for the block cipher. The parallel optimization can pre-compute the keys and IVs using combinational logic for saving these few clock cycles. As a result the clock cycles required for generation of key and IV for generation of each ciphertext are reduced.

#### 2) LIGHTWEIGHT IMPLEMENTATION

The AES algorithm consists of ten rounds and consumes ten clock cycles for implementation. However, the reduced clock cycle utilization is also secure as per the analysis of research in the past [48]. The reduced rounds proposed are also adopted by my recently proposed AE algorithms for lightweight operation [34]. Similarly, the data throughput can be increased by utilizing the less round implementation of block cipher using the same proposed algorithm for lightweight implementation. The results for implementation are discussed in the experimental results section for its applicability.

#### 3) S-BOX ARCHITECTURE IMPLEMENTATION

The proposed AE algorithm uses the S-Box for substitution for the AES-CTR algorithm as well as in the implementation for the IV generation and re-keying algorithm. The conventional method for implementation of S-Box is an implementation using the pre-generated table stored in memory. Besides, the memory, specifically SRAM, is affected mostly by the radiations in space. Therefore, the utilization of memory in space will cause a malfunction in its operation. The mitigation techniques involve the majority voter algorithm applied to memory resources, which triplicates the memory resources utilized [49]. Another approach is to avoid the usage of memory resources, tower field implementation of S-Box can be used to avoid memories [50]. This tower field optimization for the proposed AE algorithm can protect from the effects of the radiation for satellite application.

**TABLE 1. Resource utilization for implementation of proposed AE algorithm.**

| S.No. | Implementation | Area | Block-RAM | Throughput (Gbps) |
|---|---|---|---|---|
| 1 | Block Size-1 | LUT: 1296 Reg.: 806 | 10 | 6.27 |
| 2 | Block Size-2 | LUT: 1950 Reg.: 1209 | 15 | 8.70 |
| 3 | Block Size-3 | LUT: 2588 Reg.: 1612 | 20 | 10.87 |
| 4 | Block Size-4 | LUT: 3235 Reg.: 2015 | 25 | 12.76 |
| 5 | Block Size-5 | LUT: 3877 Reg.: 2418 | 30 | 14.40 |
| 6 | Block Size-6 | LUT: 4525 Reg.: 2821 | 35 | 15.86 |
| 7 | Block Size-7 | LUT: 5178 Reg.: 3224 | 40 | 17.76 |
| 8 | Block Size-8 | LUT:5825 Reg.: 3627 | 45 | 18.35 |
| 9 | Block Size-9 | LUT:6474 Reg.: 4030 | 50 | 19.43 |
| 10 | Block Size-10 | LUT: 7136 Reg.: 4433 | 55 | 20.39 |

## V. THE EXPERIMENTAL RESULTS

This section presents the implementation results obtained by the implementation of the proposed AE algorithm on FPGA. The results obtained after applying different proposed optimizations are also provided along with the comparison with the renowned AE algorithms such as AES-GCM algorithm. The experiment is performed using Xilinx Virtex-6 FPGA, and simulation is performed using Modelsim simulation software for the validation of results. The test vectors are utilized from the NIST mode of operation standard [51]. The hardware design is written in Verilog HDL. The experiment is performed using a different number of blocks of plaintext; the key size of 128-bits is utilized for experiments. The resource utilization for implementation for the different number of blocks of plaintext is shown in Table-1.

The resource utilization is calculated in the form of Look-Up Table (LUT), registers, Block-RAM (BRAM), clock frequency, and throughput. The implementation clock frequency of the proposed design on Xilinx Virtex-6 FPGA is 318.67 MHz. The throughput is calculated by multiplying the number of bits with the clock frequency of implementation and dividing the product with the number of clock cycles used for implementation. The total implementation clock cycles for a single block of plaintext with 128-bits are twelve clock cycles. Where one clock cycle is used for IV generation and re-keying, ten clock cycles used for AES block-cipher and one clock cycle to exclusive-OR of key, plaintext, and keystream. It has been noticed that by increasing the block of the plaintext of 128-bit increase one clock cycle for implementation. As the IV generation and re-keying algorithm work in chaining mode and input of the next IV are dependent on the output of the previous IV generated. Similarly, the rekeying algorithm works on the same phenomena. Hence, the generation of one ciphertext and authentication tag requires thirteen clock cycles.

The single ciphertext is produced using twelve clock cycles, combining one clock cycles for key and IV generation, ten clock cycles for ten rounds of the AES algorithm, and one clock cycle for exclusive-OR of Key, IV, and keystream. Besides, the subsequent ciphertext needs the addition of one more clock cycle for the generation of ciphertext. Therefore, the generation of ten ciphertexts involves twenty two clock cycles. The authentication tag requires one more clock cycle and total twenty three clock cycles are used for generation of ten blocks. Also, the algorithm provides 128-bits block output on every clock cycle after a delay of thirteen clock cycles. This property makes the throughput of the algorithm cumulatively to 40.78 Gbps (128∗318.67/1). Figure 3 shows the simulation results of the proposed algorithm implementation. The implementation of proposed AE algorithm is performed using same plaintext (f0f1f2f3f4f5f6f7f8f9000000000000). It also uses one key for generating more keys from the re-keying algorithm and for block cipher for the generation of ciphertext. Also, it utilizes one IV for the generation of IV's. Table 2 shows the values of the IVs and keys generated. The IV and keys generated are tested using the NIST run and frequency test, the IV's and keys have passed the run and frequency test.

### A. ARCHITECTURAL OPTIMIZATION

The different architectural optimizations proposed for the implementation of the proposed algorithm are implemented on Virtex-6 FPGA for the realization of the pros and cons of the proposed implementation.

#### 1) PARALLEL ARCHITECTURE IMPLEMENTATION

Although the architecture is parallel, the IV generation and re-keying algorithms are implemented in chaining mode. Besides, the algorithm can be utilized in full parallel mode by pre-computing the IV and keys. The pre-computation is performed using combinational logic. The pre-computation of IV and keys for re-keying will save clock cycles and provide a parallel implementation without the chaining mode of IV generation and re-keying algorithms. This optimization will increase the throughput of ten blocks of the plaintext of 128-bits from 21.28 Gbps to 40.78 Gbps. It is because the clock cycles for implementation will be ten cycles for AES block cycle one for exclusive-OR of keys, plaintexts, and key streams (128*11*318.67 /11).

The results of the resource utilization of parallel architecture optimization 1 are shown in Table-3.

#### 2) LIGHTWEIGHT IMPLEMENTATION

The lightweight functionality is desirable for many applications, including the satellite application. The increase in the trend of the development of micro and mini satellites has limited resources on-board. Instead of limiting the functionality, lightweight algorithms can provide wide applications on less onboard resources. The AES algorithm consists of ten rounds, and as per analysis in the past [48], four rounds are significant for the provision of security of data. The implementation of

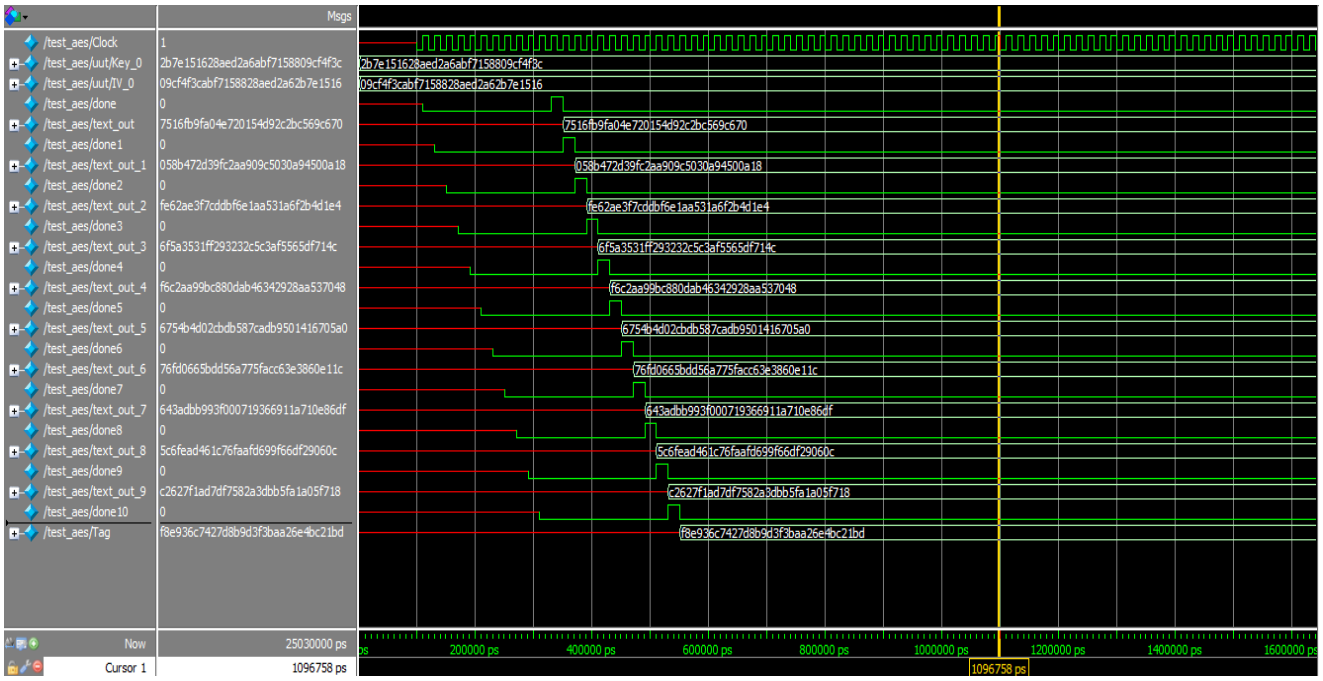


**FIGURE 3.** Simulation results of the proposed parallel nonce misuse resistant AE algorithm using Modelsim simulation software on Xilinx Virtex-6 FPGA.

**TABLE 2.** Generated IV and keys for re-keying and IV generation algorithms.

| Value | IV Generation | Re-keying |
|---|---|---|
| 0 | 09CF4F3CABF7158828AED 2A62B7E1516 | 2B7E151628AED2A6ABF7 158809CF4F3C |
| 1 | AE67054CC8EBCB389833A 0295882E3D6 | 2B82E3D69833A029C8EB CB380367054C |
| 2 | FF304A6377796CFA4BBC2 93E90092702 | 2BF227024BBC293E5D796 CFA8D9E4A63 |
| 3 | E7B56680DE81FBDB5D75 A31F537858E7 | 5813F9E75D75A31F0581F BDBAB796664 |
| 4 | 7EA4344E9CEDE171869C6 7B7D2EFF579 | 908B34E6869C67B7275EE 171F2963464 |
| 5 | 84F3FB8B1BDC39484C479 A7D1F7FA991 | E9E43C6478479A7DE89D E4486840FB64 |
| 6 | DD450DE169CC296D954EE 4767282CC36 | BE03F6C8894EE4768BCB 3C12E944A670 |
| 7 | D9C91DCBB96A4F6A16341 9261309B744 | 9FF68BA8793419265A3D BE2563C9B440 |
| 8 | 3F9A409EB23179F149CAA 71800D53DD1 | 8B85EAA389D0A718EC96 2BED399A0EF0 |
| 9 | 314EEEC3FF49D0652BB1B 4A062888FF5 | A7AF623A7EF3B4A00C57 EC662F4E9CFB |
| 10 | 5CAFACA822AD2E0748090 E97C69124E4 | EC844C4F17DE0E975B15 F7E701AF76B4 |
| 11 | 15A8AF67B6553D5F2B9A4 E7710D8CBAB | 43A673275FB79C77A19A BA293CFDB3F9 |

**TABLE 3.** Resource utilization for implementation of proposed AE algorithm with parallel architecture optimization.

| S.No. | Implementation | Area | Block-RAM | Throughput (Gbps) |
|---|---|---|---|---|
| 1 | Block Size-1 | LUT: 1296 Reg.: 806 | 10 | 7.41 |
| 2 | Block Size-2 | LUT: 1950 Reg.: 1209 | 15 | 11.12 |
| 3 | Block Size-3 | LUT: 2588 Reg.: 1612 | 20 | 14.83 |
| 4 | Block Size-4 | LUT: 3235 Reg.: 2015 | 25 | 18.54 |
| 5 | Block Size-5 | LUT: 3877 Reg.: 2418 | 30 | 22.25 |
| 6 | Block Size-6 | LUT: 4525 Reg.: 2821 | 35 | 25.96 |
| 7 | Block Size-7 | LUT: 5178 Reg.: 3224 | 40 | 29.67 |
| 8 | Block Size-8 | LUT:5825 Reg.: 3627 | 45 | 33.38 |
| 9 | Block Size-9 | LUT:6474 Reg.: 4030 | 50 | 37.09 |
| 10 | Block Size-10 | LUT: 7136 Reg.: 4433 | 55 | 40.78 |

four rounds AES algorithm is used in an algorithm known as the POET algorithm, which is presented in the CEASAR competition [34]. In this work, for the proposed AE algorithm, we also propose a lightweight implementation by using four rounds of AES algorithm. The resource utilization for this optimization is mentioned in Table-4. Besides, the data throughput is increased. The lightweight version requires six clock cycles for the implementation of a single block of 128-bits size.

Besides the authentication tag require one more clock cycle and makes the total clock cycles seven for implementation. The clock frequency remains the same as for the implementation of proposed algorithm, but the clock cycles required for lightweight implementation are reduced.

**TABLE 4.** Resource utilization for implementation of proposed AE algorithm with lightweight implementation optimization.

| S.No. | Implementation | Area | Block-RAM | Throughput (Gbps) |
|---|---|---|---|---|
| 1 | Block Size-1 | LUT: 1296 Reg.: 806 | 10 | 11.65 |
| 2 | Block Size-2 | LUT: 1950 Reg.: 1209 | 15 | 15.29 |
| 3 | Block Size-3 | LUT: 2588 Reg.: 1612 | 20 | 18.12 |
| 4 | Block Size-4 | LUT: 3235 Reg.: 2015 | 25 | 20.39 |
| 5 | Block Size-5 | LUT: 3877 Reg.: 2418 | 30 | 22.24 |
| 6 | Block Size-6 | LUT: 4525 Reg.: 2821 | 35 | 23.79 |
| 7 | Block Size-7 | LUT: 5178 Reg.: 3224 | 40 | 25.10 |
| 8 | Block Size-8 | LUT:5825 Reg.: 3627 | 45 | 26.22 |
| 9 | Block Size-9 | LUT:6474 Reg.: 4030 | 50 | 27.19 |
| 10 | Block Size-10 | LUT: 7136 Reg.: 4433 | 55 | 28.04 |

**TABLE 5.** Resource utilization for implementation of proposed AE algorithm with S-Box optimization.

| S.No. | Implementation | Area | Block-RAM | Throughput (Gbps) |
|---|---|---|---|---|
| 1 | Block Size-1 | LUT: 2829 Reg.: 1057 | 0 | 4.39 |
| 2 | Block Size-2 | LUT: 4219 Reg.: 1583 | 0 | 6.15 |
| 3 | Block Size-3 | LUT: 5621 Reg.: 2109 | 0 | 7.69 |
| 4 | Block Size-4 | LUT: 7023 Reg.: 2635 | 0 | 9.04 |
| 5 | Block Size-5 | LUT: 8425 Reg.: 3161 | 0 | 10.25 |
| 6 | Block Size-6 | LUT: 9827 Reg.: 3687 | 0 | 11.33 |
| 7 | Block Size-7 | LUT: 11229 Reg.: 4213 | 0 | 12.30 |
| 8 | Block Size-8 | LUT:12631 Reg.: 4739 | 0 | 13.18 |
| 9 | Block Size-9 | LUT:14033 Reg.: 5265 | 0 | 13.98 |
| 10 | Block Size-10 | LUT: 15458 Reg.: 5791 | 0 | 14.71 |

### 3) S-BOX ARCHITECTURE IMPLEMENTATION

The third optimization for the proposed algorithm is based on the utilization of algorithms in the space environment. The memories such as SRAM are affected by radiations and force the associated algorithm to malfunction. In the past, the tower field implementation was implemented [50]. Therefore in the proposed AE algorithm, to avoid using the memories, a tower field approach is proposed to be adopted for optimizing the performance. The tower field implementation implements the S-Box table on the logic area of FPGA instead of storing the S-Box table in memory block on FPGA. The implementation requires more area utilization in terms of Lookup tables (LUT) and Registers, but it will not utilize the memory. The resource utilization of the algorithm on the proposed algorithm is shown in Table 5. The implementation frequency of the proposed AE algorithm is decreased with optimization -3 (i.e. 240.39 MHz). Therefore the data throughput also reduced as compared to the proposed algorithm.

In addition to the individual optimizations, the combination of optimizations is very useful for optimizing the performance of proposed AE algorithms. The optimization 1 can be applied in addition to the optimization 2 for lightweight and parallel implementation of the proposed algorithm. The results of applying both optimizations (1 and 2) consumes five clock cycles for any number of the block of 128-bits size. Similarly, the optimization 2 can be used in addition to the optimization 3 for lightweight and radiation tolerant implementation.

The results of using both optimizations (2 and 3) consume seven clock cycles for a single block of 128-bits; the increase of blocks of plaintext increases the clock cycles by 1. The implementation frequency also reduced to the same frequency of S-Box optimization. Similarly, the optimization 1 can be applied in addition to the optimization 3 for parallel and radiation tolerant implementation. The results of using both optimizations (2 and 3) consume 11 clock cycles for any number of the block of 128-bits. The implementation frequency also reduced to the same frequency of S-Box optimization. Besides, the optimization 1 can be applied in addition to the optimization 2 and optimization 3 for parallel, lightweight, and radiation tolerant implementation. The results of using all optimizations (1, 2, and 3) consume five clock cycles for any number of the block of 128-bits. The implementation frequency also reduced to the same frequency of S-Box optimization. So, the optimizations are implementable alone as well as with the combinations of optimization for further improvement. The impact of individual optimization and a combination of optimizations for data throughput is shown in figure 4 and for resource utilization

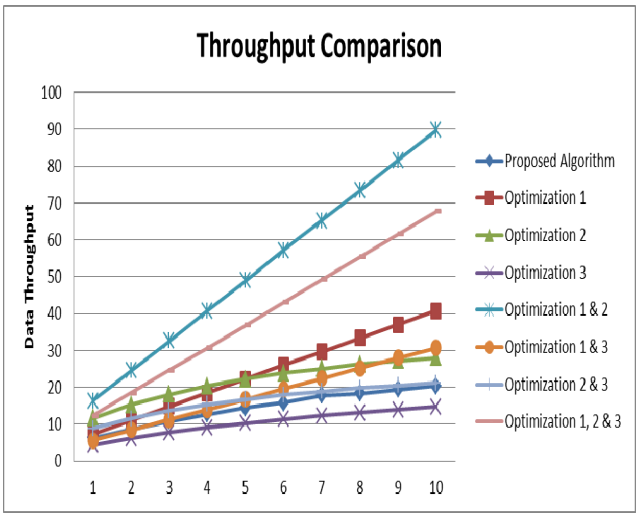


**FIGURE 4.** Graph showing the comparison of data throughput between the proposed algorithm with different optimizations and combinations of optimization.

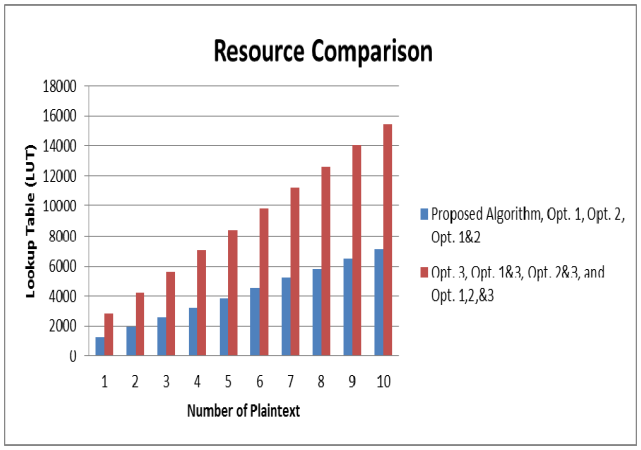


**FIGURE 5.** Graph showing the comparison of resource utilization of the proposed algorithm with different optimizations and the combination of optimization.

is shown in figure 5. Figure 4 shows different optimizations results and the combination of different optimizations results for data throughput. For ten plaintext blocks implementation, the combination of optimization 1 & 2 gives the highest throughput, followed by the combination of optimization 1, 2, and 3. Besides, the data throughput of optimization 3 has the lowest value. Figure 5 shows resource utilization for different optimizations and a combination of different optimizations. The optimization 3, combination of optimization 1 & 3, combination of optimization 2 & 3, and combination of optimization 1, 2, & 3 give high resources utilization. Besides, the optimization 1, optimization2, and combination of optimization 1 & 2 give low resource utilization. The analysis performed shows the combination of optimization 1 & 2 gives high data throughput with low resource consumption, although it does not avoid SRAM usage for radiation tolerance. Therefore, the combination of optimization 1, 2, & 3 provides better features like high throughput and radiation tolerance but consumes more hardware resources. The space environment is majorly affected by radiations; therefore, the combination of optimization 1, 2, & 3 is recommended for use in satellite applications.

### B. COMPARISON OF PROPOSED WORK

There are different AE algorithms with nonce misuse protection, such as the AES-GCM-SIV algorithm. Also, the AES-GCM algorithm can provide high throughput implementation and used in a wide variety of applications. The CCSDS and ESA standards recommended the AES-GCM algorithm for satellite application, therefore in this work, a comparison with the AES-GCM and AES-GCM-SIV is performed. In the past, the implementation of the AES-GCM algorithm on FPGA is presented by B. Yang *et al.* [27]. The implementation provides pipeline optimization for the increase in data throughput. S. Lemsitzer *et al.* [29] have presented his AES-GCM implementation with an emphasis on AES implementation optimization using Xilinx Virtex-4 FPGA. Recently, J. Vliegen *et al.* [30] showed an AES-GCM algorithm implementation on Xilinx Virtex 7 FPGA with DPA attack resistant. G. Zhou *et al.* [53] implemented his rapid implementation on Xilinx FPGA using the karatsuba multiplier. L. Henzen *et al.* [54] have presented his 100 Gbps implementation of the AES-GCM algorithm for Ethernet application on Xilinx Virtex 5 FPGA. K. M. Abdellatif *et al.* [55], [56] has shown his high-speed implementation for slow-changing key application and AEGIS algorithm using Xilinx Virtex 5 FPGA. Besides, Y. Zhang *et al.* [52] have presented the high throughput implementation of the AES-GCM algorithm using pipeline architecture on Xilinx Virtex-5 FPGA. Koteshwara *et al.* [12] implemented the AES-GCM and AES-GCM-SIV and compared their resources along with optimizations on Altera Cyclone V FPGA.

The proposed AE algorithm resource utilization, along with the two recommended optimizations for satellite application, is compared with previous implementations. The proposed AE algorithm with optimization not only protects against the side-channel and nonce misuse attacks but provides high data throughput and better efficiency. Table-6 provides the comparison of area utilization as well as the data throughput comparison of the proposed algorithm with different implementations.

## VI. THE SECURITY ANALYSIS AND DISCUSSION

The proposed AE algorithm consists of three parts IV generation, re-keying, and AES-CTR algorithms. The algorithm responsible for the confidentiality of the proposed AE algorithm is the AES-CTR algorithm; many researchers have provided the AES-CTR algorithm security analysis for ten rounds [57]–[59] and four rounds [60], [61]. Besides the security of AES-CTR depends on the IV, the IV must not be repeated as using the same key, and IV pair can generate the same ciphertext. Therefore the most critical inputs for the AES-CTR algorithm are the key and IV; therefore, in the proposed algorithm, an IV generation algorithm along with a re-keying algorithm is used for the randomization of both vector for enhancing security against linear and differential attacks.

### A. BRUTE FORCE AND FREQUENCY OF LETTER ATTACK

The brute force attack involves trying the possible number of combinations of key and IV for breaking the security of the AES-CTR algorithm. The generic AES-CTR algorithm requires $2^{128}$ combinations of IV along with $2^{128}$ combinations of plaintext, which make $2^{256}$ combinations for a brute force attack to hold with a constant key.

In the proposed AE algorithm, the IV is randomized, and the re-keying algorithm generates a randomized key for each ciphertext. Therefore, for brute force attack, the $2^{128}$ combination of IV, $2^{128}$ combination of the key, and $2^{128}$ combination of plaintext will make a total of $2^{384}$ combinations with constant key for the block cipher.

**TABLE 6.** Resource comparison with AES-GCM and AES-GCM-SIV algorithms implementations.

| Implementation | Clock freq. (MHz) | Area | Block-RAM | Throughput (Gbps) | Efficiency (Mbps/Area) |
|---|---|---|---|---|---|
| B. Yang [27] | 271 | 463328 | 0 | 34.7 | 0.07 |
| S. Lemsitzer [29] | 120 | 27800 | 0 | 15.3 | 0.55 |
| J. Vliegen [30] | 119 | 38241 | 0 | 15.2 | 0.39 |
| G. Zhou [53] | 305 | 8077 | 0 | 39.0 | 4.83 |
| L. Henzen [54] | 233 | 18505 | 0 | 48.8 | 2.63 |
| K. M. Abdellatif [55] | 264 | 7475 | 0 | 31.4 | 4.19 |
| K. M. Abdellatif [56] | 232 | 5512 | 0 | 29.7 | 5.38 |
| Y. Zhang [52] | 381 | 6482 | 0 | 48.8 | 7.54 |
| Koteshwara GCM [12] | 50 | 4087 | 0 | 4.1 | 1.02 |
| Koteshwara GCM-SIV [12] | 50 | 4262 | 0 | 3.9 | 0.93 |
| Proposed Method | 318 | 7136 | 55 | 20.3 | 2.85 |
| Optimization 1 & 2 | 318 | 7136 | 55 | 89.7 | 12.57 |
| Optimization 1, 2, & 3 | 240 | 15458 | 0 | 67.6 | 4.37 |

**TABLE 7.** NIST frequency and run test on IV and key.

| Blocks | Frequency IV | Frequency Key | RUN IV | RUN Key |
|---|---|---|---|---|
| 1 | 0.5959 | 0.3768 | 0.8399 | 2.0000 |
| 2 | 0.4795 | 0.1573 | 1.6028 | 1.8940 |
| 3 | 0.4795 | 0.8597 | 0.6897 | 0.1109 |
| 4 | 0.7237 | 0.1573 | 0.0750 | 0.1508 |
| 5 | 0.7237 | 0.5959 | 0.1088 | 0.0199 |
| 6 | 0.5959 | 0.7237 | 0.0724 | 0.0503 |
| 7 | 0.2888 | 0.5959 | 1.2023 | 1.9773 |
| 8 | 0.8597 | 0.3768 | 1.6219 | 0.1356 |

The frequency of letter attack involves trying the relationship between the ciphertext letters frequency for guessing the plaintext for breaking the security of the AES-CTR algorithm. In the proposed AE algorithm, the IV is more randomized as compared to the generic algorithm, and the re-keying algorithm generates a more randomized key as well for each ciphertext. The test of the frequency of letter attack is performed using the NIST's statistical tests for randomness such as run and frequency tests [62]. The results of the NIST run and frequency test on the generated keys by re-keying algorithm and IV using the IV generation algorithm are shown in Table 7. The IV and key generated using the proposed algorithm has passed the NIST run and frequency test.

The re-keying algorithm was designed to generate a new session key for every ciphertext generated using key and IV combination. The probability of using the same key with the same IV and plaintext is very low; therefore, the key is secure and as OTP. So, we can see that even if a key is repeated, the chances of the same corresponding IV is negligible.

### B. PRE-COMPUTATION ATTACK

In addition to the brute force attack, there are two more effective attacks based on the pre-computation of a large database before attacking the algorithm. These attacks are a key collision attack and Hellman's time-memory tradeoff attack. These attacks provide a shorter time as compared to a brute force attack. Besides, more memory is required for pre-computation. The attacks do not require any knowledge of plaintext during the pre-computation. In a key collision attack, a number of bits of secret keys are n. the key collision attack efficiency can guess the key size of n-lgM, where M is the number of secret keys. On the other hand, Hellman's attack has a crucial adequate size of 2n/3. The proposed AE algorithm has an efficiency of 256-lgM for key collision and 170 bits for Hellman's attack, as compared to the 128-lgM for key collision and 85 for Hellman's attack for AES-CTR algorithm.

However, a block cipher can protect the pre-computation attacks, the dependence of AES-CTR on IV significantly effective against the pre-computation attacks. The attacker computes the value using IV in the pre-computation stage, and by randomizing the IV, create the computational difficulties for the attacker twofold. Also, by initializing the IV with a secret vector or using an unpredictable generating algorithm can protect against the pre-computation attack. A similar approach is integrated into the proposed AE algorithm by using a secret IV initially the same as a key. Hence, the pre-computation attacks are ineffective against the proposed AE algorithm.

### C. KNOWN OR CHOSEN PLAINTEXT ATTACK

Let us suppose; the attacker has plaintext known which are P1and P2 encrypted using key K1. Then the generated ciphertext using the AES-CTR generic algorithm is

(P1 XOR K1), (P2 XOR K1)
The attackers can XOR the selected sequence S1 and S2 to get the required information.
(S1 XOR (P1 XOR K1)), (S2 XOR (P2 XOR K1))
(S1 XOR P1)XOR K1), (S2 XOR P2) XOR K1)
(S1 XOR P1), (S2 XOR P2)
Although the proposed AE algorithm also has re-keying unique key XOR with the plaintext, so it will make it difficult for the attacker to break the security.
(S1 XOR (P1 XOR K1 XOR Ki)), (S2 XOR (P2 XOR K1 XOR Ki))

(S1 XOR P1 XOR Ki)XOR K1), (S2 XOR P2 XOR Ki) XOR K1)
(S1 XOR P1 XOR Ki), (S2 XOR P2 XOR Ki)
Hence the proposed AE algorithm is secure against the known or chosen-plaintext attack.

### D. KNOWN OR COMPROMISED KEY ATTACK

Let us suppose the attacker knows the value of the original key, so by knowing the key expansion algorithm, he can guess the generated keys for the AES-CTR algorithm. But on the contrary, the attacker does not know the initial secret IV, so the re-keying algorithm output and IV generation algorithm output is not known to him, which intern makes the security of the proposed AE algorithm intact. Besides, the attacker does not know the keys from the rekeying algorithm, which makes the attack weak as by knowing the key, the attacker cannot guess the re-keying keys using the ciphertext. The $2^{128}$ combination of keys from the re-keying algorithm and $2^{128}$ combination of IV with a total $2^{256}$ combination of input values are unknown to the attacker. Hence, by knowing a single key will not help the attacker to guess the plaintext. Whereas the proposed algorithm uses the key in IV generation and IV in the key generation makes the problems for attack twofold.

### E. SIDE CHANNEL ATTACK

The side-channel attacks are attacks launched on the electromagnetic pattern of transmitted ciphertext or the transmitted power analysis of ciphertext such as DPA attack. Although there are many side-channel attacks apart from the two methods discussed, the mentioned methods are the most effective side-channel attacks carried out. The strength of the side-channel attacks is based on the pattern repetition to guess the similarity of the pattern. The proposed AE algorithm has randomized the IV and keys from a re-keying algorithm which makes the ciphertext to become random as different from other generated ciphertext. Therefore, the side-channel attacks, such as the DPA attack, will not be feasible for breaking the security of the proposed AE algorithm. The phenomena of the provision of new session keys are utilized in the proposed AE algorithm, which is the essential requirement for avoiding side-channel attacks.

### F. NONCE MISUE ATTACK

The nonce misuse attack includes the use of the same nonce in the IV for the generation of ciphertext. In the generic AES-CTR algorithm, the increment operation is performed for generating a unique IV for a ciphertext. However, still, the nonce remains the same, and only the counter field is incremented. The proposed AE algorithm uses the secret IV for initial IV, and the later IVs are generated using the IV generation algorithm. The IV generation algorithm makes IV's random for generating the ciphertext. Also, the same nonce is not utilized in the IV for generating the ciphertext. Therefore, the nonce misuse attack is not effective in the proposed AE algorithm. Also, the counter field overflow fault for IV repetition that exists in the generic algorithm does not apply to the proposed AE algorithm. Also, the replay attack, such as sending the ciphertext again to the proposed AE algorithm will not work as the key and IV are changed every time.

## VII. CONCLUSION

This work proposed a parallel AE algorithm with nonce misuse protection and side-channel attack resistant features. The proposed algorithm uses the IV generation algorithm for nonce misuse protection by randomization of secret IV. Also, the side channels attacks such as DPA attack is ineffective on algorithm because of using the re-keying algorithm. Meanwhile, three optimizations are also proposed for achieving high data throughput, lightweight implementation, and space radiations protection. The effects of the combination of the optimizations on hardware implementation are also discussed. The proposed algorithm is compared with the previous implementation of AES-GCM and AES-GCM-SIV algorithms for comparing the effectiveness of the proposed algorithm. The security analysis of the proposed algorithm is presented for validation of the proposed algorithm.

## REFERENCES

[1] UN-SPIDER. *New Chinese Satellites Will Support the Country's Disaster Management*. Accessed: Nov. 1, 2019. [Online]. Available: http://www.un-spider.org/newsand-events/news/new-chinese-satellites-will-support-disastermanagement
[2] S. J. H. Pirzada, A. Murtaza, T. Xu, and L. Jianwei, "Disaster management using IP-based space-air-ground information network," in *Proc. IEEE Int. Conf. Unmanned Syst. (ICUS)*, Beijing, China, 2019, pp. 119–123.
[3] A. Murtaza, S. J. Hussain Pirzada, L. Jianwei, and T. Xu, "Air traffic surveillance using IP-based space information network," in *Proc. 28th Wireless Opt. Commun. Conf. (WOCC)*, Beijing, China, May 2019, pp. 1–6.
[4] CCSDS 350.0-G-2. (Jan. 2006). *The Application of CCSDS Protocols to Secure Systems*. [Online]. Available: https://public.ccsds.org/Pubs/350x0g2.pdf
[5] M. Dworkin, *Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality*, document 800-38C, NIST, 2007.
[6] *Information Technology—Security Techniques—Authenticated Encryption*, Standard ISO/IEC 19772:2009, International Organization for Standardization, 2009.
[7] D. A. McGrew and J. Viega, "The security and performance of the Galois/Counter Mode (GCM) of operation," in *Proc. 5th Int. Conf. Cryptol. IndiaSecur*, 2004, pp. 343–355.
[8] CCSDS 352.0-B-1. (Nov. 2012). *CCSDS Cryptographic Algorithms*. [Online]. Available: https://public.ccsds.org/Pubs/352x0b1.pdf
[9] B. H. A. Zauner, S. Devlin, J. Somorovsky, and P. Jovanovic, "Nonce-disrespecting adversaries: Practical forgery attacks on GCM in TLS," in *Proc. USENIX WOOT*, 2016, pp. 1–11.
[10] D. Harkins, *Synthetic Initialization Vector (SIV) Authenticated Encryption Using the Advanced Encryption Standard (AES)*, document RFC 5297, Oct. 2008.
[11] S. Gueron, A. Langley, and Y. Lindell, *AES-GCM-SIV: Nonce Misuse-Resistant Authenticated Encryption*, document RFC 8452, Apr. 2019. Accessed: Nov. 1, 2019.
[12] S. Koteshwara, A. Das, and K. K. Parhi, "Performance comparison of AES-GCM-SIV and AES-GCM algorithms for authenticated encryption on FPGA platforms," in *Proc. 51st Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, Oct. 2017, pp. 1331–1336.
[13] CAESAR. *Competition for Authenticated Encryption: Security, Applicability, and Robustness*. [Online]. Available: http://competitions.cr.yp.to/caesar.html/

[14] S. Koteshwara and A. Das, "Comparative study of authenticated encryption targeting lightweight IoT applications," *IEEE Des. Test. IEEE Des. Test. Comput.*, vol. 34, no. 4, pp. 26–33, Aug. 2017.
[15] T. Shi, C. Jin, and J. Guan, "Collision attacks against AEZ-PRF for authenticated encryption AEZ," *China Commun.*, vol. 15, no. 2, pp. 46–53, Feb. 2018.
[16] *PRIMATEs v1.02.*, CAESAR, Rome, Italy, 2014.
[17] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," *Lect. Notes Comput. Sci.*, vol. 2004, vol. 3156, pp. 16–29.
[18] P. Kocher, J. Jaffe, B. Jun, and P. Rohatgi, "Introduction to differential power analysis," *J. Cryptogr. Eng.*, vol. 2011, vol. 1, no. 1, pp. 5–27.
[19] M. Zaba, N. Jamil, M. Rohmad, H. Abdul, and S. Shamsuddin, "The CiliPadi family of lightweight authenticated encryption (version 1)," NIST Lightweight Cryptogr. Round-1, Gaithersburg, MD, USA, 2019.
[20] E. Prouff and M. Rivain, "Masking against side-channel attacks: A formal security proof," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 7881. Heidelberg, Germany: Springer, 2013, pp. 142–159.
[21] A. C. Dobraunig, F. Koeune, S. Mangard, F. Mendel, and F.-X. Standaert, "Towards fresh and hybrid re-keying schemes with beyond birthday security," in *Smart Card Research and Advanced Applications* (Lecture Notes in Computer Science), vol. 9514. Heidelberg, Germany: Springer, 2015, pp. 225–241.
[22] C. Dobraunig, M. Eichlseder, S. Mangard, F. Mendel, and T. Unterluggauer, "ISAP—Towards side-channel secure authenticated encryption," *IACR Trans. Symmetric Cryptol.*, vol. 2017, no. 1, pp. 80-105, 2017.
[23] F. Brosser, E. Milh, V. Geijer, and P. Larsson-Edefors, "Assessing scrubbing techniques for xilinx SRAM-based FPGAs in space applications," in *Proc. Int. Conf. Field-Program. Technol. (FPT)*, Shanghai, China, Dec. 2014, pp. 296–299.
[24] C. I. Underwood and M. K. Oldfield, "Observations on the reliability of cots-device-based solid state data recorders operating in low Earth orbit," in *Proc. 5th Eur. Conf. Radiat. Effects Compon. Syst.*, Fontevraud, France, 1999, pp. 387–393.
[25] J. Wilkinson and S. Hareland, "A cautionary tale of soft errors induced by SRAM packaging materials," *IEEE Trans. Device Mater. Rel.*, vol. 5, no. 3, pp. 428–433, Sep. 2005.
[26] S. J. H. Pirzada, S. of Cyber Science, China. TechnologyBeihang UniversityBeijing, A. Murtaza, L. Jianwei, and T. Xu, "Single event effects tolerant AES-CTR implementation for authentication of satellite communication," *Int. J. Comput. Commun. Eng.*, vol. 8, no. 4, pp. 178–183, 2019.
[27] B. Yang, S. Mishra, and R. Karri, "A high-speed architecture for Galois/Counter Mode of operation (GCM)," in *Proc. IACR Cryptol. ePrint Arch.*, 2005, p. 146.
[28] J. C. Resende and R. Chaves, "Compact dual block AES core on FPGA for CCM protocol," in *Proc. 25th Int. Conf. Field Program. Log. Appl. (FPL)*, Sep. 2015, pp. 1–8.
[29] S. Lemsitzer, J. Wolkerstorfer, N. Felber, and M. Braendi, "Multi-gigabit GCM-AES architecture optimized for FPGAs," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst. (CHES)*, 2007, pp. 227–238.
[30] J. Vliegen, O. Reparaz, and N. Mentens, "Maximizing the throughput of threshold-protected AES-GCM implementations on FPGA," in *Proc. IEEE 2nd Int. Verification Secur. Workshop (IVSW)*, Thessaloniki, Greece, Jul. 2017, pp. 140–145.
[31] Y. Wang and Y. Ha, "High throughput and resource-efficient AES encryption/decryption for sans," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2016, pp. 1166–1169.
[32] S. J. Hussain Pirzada, A. Murtaza, L. Jianwei, and T. Xu, "The parallel CMAC authenticated encryption algorithm for satellite communication," in *Proc. IEEE 9th Int. Conf. Electron. Inf. Emergency Commun. (ICEIEC)*, Beijing, China, Jul. 2019, pp. 1–5.
[33] S. J. H. Pirzada, A. Murtaza, M. N. Hasan, T. Xu, and L. Jianwei, "The parallel CMAC synthetic initialization vector algorithm implementation on FPGA," in *Proc. 2nd Int. Conf. Latest Trends Electr. Eng. Comput. Technol. (INTELLECT)*, Karachi, Pakistan, Nov. 2019, pp. 1–5.
[34] *POET*, CAESAR, Rome, Italy, 2014.
[35] S. Koteshwara, A. Das, and K. K. Parhi, "FPGA implementation and comparison of AES-GCM and deoxys authenticated encryption schemes," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2017, pp. 1–4.
[36] E.-H. Bensikaddour, Y. Bentoutou, and N. Taleb, "Satellite image encryption method based on AES-CTR algorithm and GEFFE generator," in *Proc. 8th Int. Conf. Recent Adv. Space Technol. (RAST)*, Istanbul, Turkey, Jun. 2017, pp. 247–252.
[37] A. S. Bader and A. M. Sagheer, "Modification on AES-GCM to increment ciphertext randomness," *Int. J. Math. Sci. Comput.*, vol. 4, no. 4, pp. 34–40, Nov. 2018.
[38] M. S. E. Mohamed, S. Bulygin, M. Zohner, A. Heuser, M. Walter, and J. Buchmann, "Improved algebraic side-channel attack on AES," *J. Cryptograph. Eng.*, vol. 3, no. 3, pp. 139–156, Apr. 2013.
[39] C. Dobraunig, M. Eichlseder, S. Mangard, F. Mendel, and T. Unterluggauer, "ISAP-towards side-channel secure authenticated encryption," *IACR Trans. Symmetric Cryptol.*, vol. 2017, no. 1, pp. 80–105, 2017.
[40] S. Koteshwara, A. Das, and K. K. Parhi, "Architecture optimization and performance comparison of Nonce-Misuse-Resistant authenticated encryption algorithms," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 5, pp. 1053–1066, May 2019.
[41] *IEEE Standard for Verilog Hardware Description Language*, Standard IEEE Std 1364-2005 and Revision of IEEE Std 1364-2001, Apr. 2006, pp. 1–590.
[42] *IEC/IEEE International Standard—Behavioural Languages—Part 1-1: VHDL Language Reference Manual*, Standard IEC 61691-1-1:2011(E) IEEE Std 1076-2008, May 2011, pp. 1–648.
[43] S. J. H. Pirzada, S. of Cyber Science, C. TechnologyBeihang University-Beijing, A. Murtaza, T. Xu, and L. Jianwei, "The compatibility analysis of AES algorithm for design portability on FPGA," *Int. J. Comput. Theory Eng.*, vol. 11, no. 6, pp. 112–115, 2019.
[44] *Symphony High-Level Synthesis Tool (SHLS)*. Accessed: Nov. 1, 2019. [Online]. Available: https://www.microsemi.com/product-directory/dev-tools/4899-synphony
[45] Xilinx. *Embedded System Tools Reference Manual*. Accessed: Nov. 1, 2019. [Online]. Available: http://www.xilinx.com
[46] S. Chhabra and K. Lata, "Hardware-software co-simulation of obfuscated 128-bit AES algorithm for image processing applications," in *Proc. IEEE Int. Symp. Smart Electron. Syst. (ISES)*, Hyderabad, India, Dec. 2018, pp. 191–194.
[47] S. J. H. Pirzada, A. Murtaza, T. Xu, and L. Jianwei, "A reconfigurable model-based design for rapid prototyping on FPGA," *Int. J. Comput. Theory Eng.*, to be published.
[48] J. Daemen, M. Lamberger, N. Pramstaller, V. Rijmen, and F. Vercauteren, "Computational aspects of the expected differential probability of 4-round AES and AES-like ciphers," *Computing*, vol. 85, nos. 1–2, pp. 85–104, May 2009.
[49] Y. Jiang, J. Han, X. Zhu, and M. Cai, "Single event upset mitigation testing of SRAM-based FPGAs," in *J. Beijing Univ. Aeronaut. Astronaut.*, pp. 44–53, Aug. 2014.
[50] D. Canright, "A very compact S-box for AES," in *Proc. Int. Workshop Cryptograph. Hardware Embedded Syst.*, 2005, pp. 441–455.
[51] NIST, Special Publication 800-38A. (Dec. 2001). *Recommendation for Block Cipher Modes of Operation: Methods and Techniques*. [Online]. Available: https://csrc.nist.gov/publications/detail/sp/800-38a/final
[52] Y. Zhang, N. Wu, F. Zhou, X. Zhang, and J. Zhang, "High-performance AES-GCM implementation based on efficient AES and FR-KOA multiplier," *Trans. IEICE Electron. Express*, vol. 15, no. 14, Jul. 2018, Art. no. 20180559.
[53] G. Zhou, H. Michalik, and L. Hinsenkamp, "Improving throughput of AES-GCM with pipelined karatsuba multipliers on FPGAs," in *Reconfigurable Computing: Architectures, Tools and Applications* (Lecture Notes in Computer Science), vol. 5453. Berlin, Germany: Springer, 2009, pp. 193–203.
[54] L. Henzen and W. Fichtner, "FPGA parallel-pipelined AES-GCM core for 100G Ethernet applications," in *Proc. ESSCIRC*, Sep. 2010, pp. 202–205.
[55] K. M. Abdellatif, R. Chotin-Avot, and H. Mehrez, "High speed authenticated encryption for slow changing key applications using reconfigurable devices," in *Proc. IEEE Wireless Days*, Nov. 2013, pp. 1–6.
[56] K. M. Abdellatif, R. Chotin-Avot, and H. Mehrez, "AES-GCM and AEGIS: Efficient and high-speed hardware implementations," *Trans. J. Signal Process. Syst.*, vol. 88, no. 1, pp. 1–12, 2016.
[57] D. A. McGrew. (Nov. 2002). *Counter Mode Security: Analysis and Recommendations*. Accessed: Nov. 1, 2019. [Online]. Available: http://www.mindspring.com/-dmcgrew/ctr-security.pdf
[58] L. Xian and W. Tingthanathikul, "Advanced Encryption Standard (AES) in counter mode," ECE 575 Course Project, Winter '04. 1. [Online]. Available: https://pdf.Semanticscholar.org/7e2f/3c9060256bd7b0a4749cf6032736037f5.pdf
[59] F. Sibleyras. (2017). *Cryptanalysis of the Counter Mode of Operation*. [Online]. Available: https://hal.inria.fr/hal-01662040

[60] S. Park, S. H. Sung, S. Chee, E.-J. Yoon, and J. Lim, "On the security of Rijndael-like structures against differential and linear cryptanalysis," in *Advances in Cryptology— ASIACRYPT* (Lecture Notes in Computer Science), vol. 2501. Y. Zheng, Ed. Berlin, Germany: Springer, 2002, pp. 176–191.
[61] L. Keliher, "Refined analysis of bounds related to linear and differential cryptanalysis for the AES," in *Advanced Encryption Standard* (Lecture Notes in Computer Science), vol. 3373, H. Dobbertin, V. Rijmen, and A. Sowa, Eds. Berlin, Germany: Springer, 2004, pp. 42–57.
[62] A. Rukhin, J. Soto, and J. Nechvatal, *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, document 800-22, NIST, Gaithersburg, MD, USA, 2010.

**SYED JAHANZEB HUSSAIN PIRZADA** was born in Attock, Pakistan. He received the B.Eng. degree in electronics engineering from the NED University of Engineering and Technology, Karachi, Pakistan, in 2007, and the M.Sc. degree in electrical, electronics, control and instrumentation engineering from Hanyang University, Seoul, South Korea, in 2012. He is currently pursuing the Ph.D. degree with the School of Cyber Science and Technology, Beihang University, Beijing, China. He has over twelve year working experience in the National Space Agency of Pakistan. He has published more than 30 research papers in international conferences and journals. He has one patent with patent office, South Korea. He has delivered many keynote speeches and tutorial in international conferences on the topic of digital system design. His research interests are cryptography for satellite applications and framework design for space air ground information networks.

**ABID MURTAZA** was born in Karachi, Pakistan. He received the M.Sc. degree in electronics from the University of Karachi, Karachi, Pakistan, in 2010. He is currently pursuing the Ph.D. degree in space technology applications with Beihang University, Beijing, China. He has been working with the Pakistan's National Space Agency SUPARCO, since 2010. His research interests include space information networks, information security, cryptography, security protocols, and satellite communication.

**TONGGE XU** received the master's degree in engineering from the Beijing University of Aeronautics and Astronautics, in 1993. He is currently an Associate Professor with the School of Cyber Science and Technology, Beihang University, Beijing, China. His research areas are network management and flow/protocol analysis technology, UNIX/Linux system development, large information system design and development technology, as well as public opinion big data analysis and mining.

**LIU JIANWEI** was born in Shandong, China. He received the B.S. and M.S. degrees in electronics and information engineering from Shandong University, Shandong, in 1985 and 1988, respectively, and the Ph.D. degree in electronics and communication systems from Xidian University, Shaanxi, China, in 1998. He is currently the Dean of the School of Cyber Science and Technology, Beihang University, Beijing, China. His current research interests include wireless communication networks, cryptography, and information and network security.