ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 1
www.ijisea.org

Design and Implementation of RSA Cryptography Using an Inexact Multiplier

[1]K Jyoshna, [2]L Swathi
Annamacharya Institute of Technology and Sciences, Kadapa, A.P, India
Kurguntlajyoshna@Gmail.Com[1], Lingamguntaswathi@Gmail.Com[2]

ABSTARCT

Cryptography used in encryption and decryption for the purpose of security in communication. Rivest Shamir Adleman or RSA is an algorithm which is used to send data. In this, mainly focuses on encryption techniques like cryptography which have large integer multiplications like modular exponentiation. However, these operations required large hardware architecture. This makes traditional RSA implementations inefficient for low power and resource constrained applications like Internet of things (IOT) devices, embedded systems and real time multimedia encryption. To overcome these limitations, this project introduces an optimized RSA cryptographic architecture that contains 8-bit Dadda multiplier and 5:2 compressor which will balance the performance. The 5:2 compressor reduces the number of addition stages in the product multiplication, while the Dadda multiplier focuses on fast multiplication. This inexact multiplier design significantly reduces the power consumption, Low area requirement, Improved Scalability and Faster computation which is well suited for energy aware hardware environments. The proposed design achieves 27% power reduction while preserving encryption accuracy. Although there is a slight decryption error that is negligible. The encryption simulation shows that the result is unaffected. The traditional RSA design uses Exact multipliers like carry save and array-based designs which consume more power and the logic structure is complex. In contrast, the approximate computing approach sacrifices a minimal amount of accuracy in Favor of Dadda-based multipliers, offers several distinct advantages over traditional RSA hardware implementations.
Keywords: Cryptography, Dadda Multiplier, Speed, Power Consumption

1. INTRODUCTION

The main technique used in RSA Cryptography is modular exponentiation, which perform large or huge integer multiplications. When it is implemented in hardware the result is longer delays, high power consumption and more silicon space. The platforms which have limited resources like IOT nodes, embedded processors, smart cards and portable communication modules which take stringent power, speed restrictions and area, this are the serious difficulties. conventional RSA architectures use exact multipliers like array based or carry save designs. Approximate computing has came into existence to address these issues. Unlike exact multipliers it delibertely tolerates minor inaccuracies to achieve considerable improvements in power, speed, efficiency and circuit compactness. This approach has effective in domains like neural networks, multimedia and lightweight cryptography. Among different techniques, Dadda multiplier is combined with 5:2 compressor which is very effective. The Dadda tree minimize the reduction stages while 5:2 compressor optiize multi operand addition. Addition of these two will get the result of shorten critical path delay, reduce logic depth and lower energy consumption without affecting the performance. These improvements is valuable for RSA to strengthen security, placing even greater demands on modular exponentiation and multiplication.

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 1
www.ijisea.org

Approximate computing done these core operations resulting in improving both throughput and energy efficiency. Small inaccuracies caan be tolerated in cryptographic systems. A partial balance between speed, power, and area is achieved by RSA hardware through the integration of Dadda multipliers with inexact compressors. This design provides good performance for lightweight cryptography making it well suited for embedded and real time applications.

The Figure 1.1 shows the representation of optimized RSA hardware engine which uses inexact multiplication to improve the performance in terms of hardware usage and energy efficiency. Each and every block is important for encryption and decryption procedures. The first block of the hardware engine is key generation unit, which is responsible for creating the core cryptographic keys such as the public key (e), the private key (d) and modulus (n). This block is foundation of hardware engine which secure communication by generating large prime numbers and computing the necessary parameters for RSA.

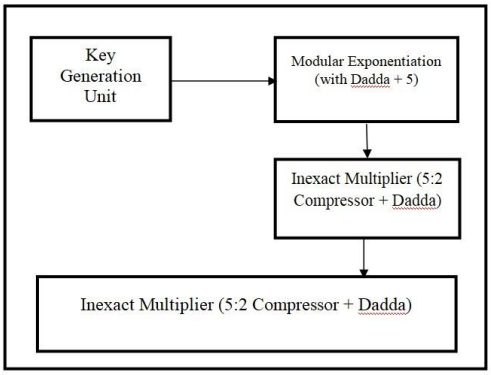


Fig 1.1: Block Diagram of RSA Hardware Engine

Fig 1.1 shows the block diagram of RSA which is using an inexact multiplier. After completing the key generation unit, the design moves to modular exponentiation unit, which performs larger task of raising the message t the power of the key (either e or d) and reducing it modulo (n). This operation is very important for both encryption and decryption processes. To optimize this block, the design uses a custom multiplier built using Dadda tree structure with 5:2 compressor. It performs multiplication while reducing the power and area. After that there are two layers of Inexact multiplier blocks, which reused the components that integrate the 5:2 compressor logic within the Dadda multiplier framework. These blocks achieve gains in latency, power consumption and chip area making them ideal for deployment in IOT devices, embedded systems and mobile security modules.

2. LITERATURE REVIEW

The modified two stage 5:2 compressors have analyzed performance metrics by using built in multipliers [1]. They have proposed upon new 4:2 compressor using two optimizes approximate unsigned multipliers, giving better tradeoffs for low power systems [2]. These team focused on energy efficiency and computational simplicity by analyzing approximate adders and multipliers based on majority logic [3]. Presented at the IEEE nanotechnology conference with introduced 6:2 majority logic based imprecise compressor intended for use in approximate multipliers [4]. Balancing energy efficiency with acceptable imprecision by using compressor designs optimized nanoscale technology [5]. Conducted a comparative study of FIR filter architectures that leverages distributed arithmetic for improved performance. Aiming to enhance

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 1
www.ijisea.org

approximation efficiency using the implementation of majority logic in the design of arithmetic circuits. The system reliability examines how approximate computing poses both challenges and opportunities for cyber security
[6-8].

3. METHODOLOGY

3.1 Existing Method

The current landscape of cryptographic hardware design primarily revolves around traditional RSA implementations, which are highly dependent on exact multipliers such as modular exponentiation and multiplications. These operations are computationally intensive which require complex algorithms to handle large integers, especially public key cryptography. The main challenge in implementing RSA on hardware is the tradeoff between performance and hardware resource utilization. Conventional designs like carry save or array-based multipliers provide accurate results but demand more logic resources resulting in increase of power consumption, latency and silicon area.

In many existing RSA hardware systems, the computation is dominated by modular multiplication and exponentiation however it is critical to the encryption and decryption operations. These systems use exact multipliers such as carry save and array-based designs and compressors to handle the partial production generation and addition steps. This gives perfect precision, but the amount of power consumption and area is increased. The standard multipliers which are used in most of the common implementations, such as those based on the Wallace tree or the Dadda tree. The methods are not optimal for low power or space-constrained applications.

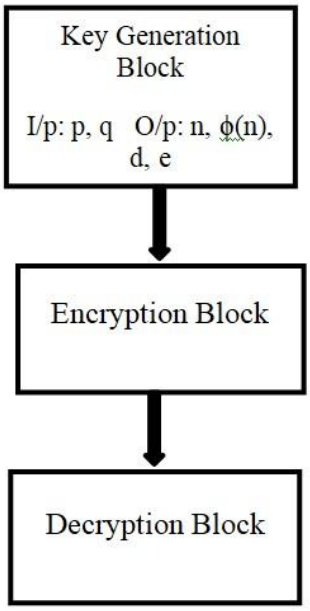


Fig 3.1: Flow chart diagram of RSA

In RSA the modular exponentiation is expressed as C = Me mod n for encryption process and M = Cd mod n for decryption process, which requires repeated large number multiplications. The figure shows the flow

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 1
www.ijisea.org

chart diagram of RSA which has three steps key generation unit, Encryption block and decryption block. The conventional RSA architectures multiplications are done by exact multipliers, which include full adders, carry save adders and ripple carry adders.

The reliance on exact multipliers ensures correctness but at the cost of high-power consumption, slower performance, and greater complexity. Consequently, conventional RSA architecture, while secure, is poorly suited for low-power and resource-constrained hardware applications. These designs give the perfect accuracy but known to suffer from High power consumption, larger silicon area and longer propagation delays. These are not ideal for resource constrained platforms like IOT devices and embedded systems.

3.2 Proposed Method

The Dadda multiplier is a high-speed column compression multiplier designed to minimize logic levels during partial product reduction. Unlike traditional array or Wallace multipliers, it generates partial products and compresses them in a tree-like manner, resulting in lower propagation delay. In this work, a Dadda tree is combined with a 5:2 compressor, which can process five input bits and two carry-ins with low latency.

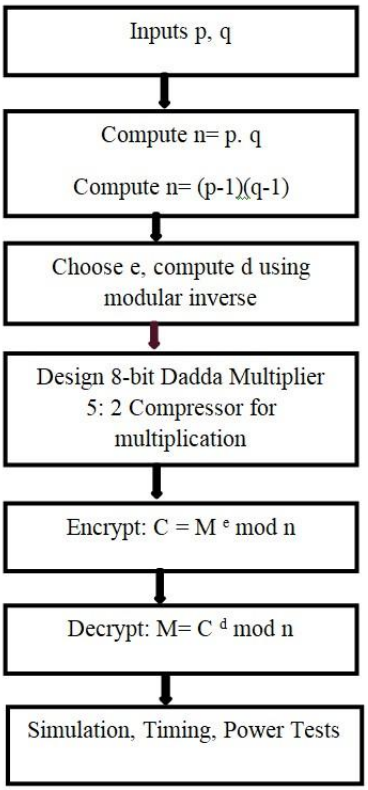


Fig 3.2: RSA Computation Block with Inexact Multiplier

By applying approximate computing to these multiplications, significant gains in speed and power efficiency are achieved. While encryption is largely unaffected by minor approximation errors, decryption may show slight deviations. The design adopts a modular approach, with separate blocks for key generation, encryption, decryption, and the inexact multiplier. Simulation verifies correct functionality, and synthesis reports indicate reduced LUT utilization, slice usage, and overall power consumption. This design delivers a hardware

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 1
www.ijisea.org

efficient and energy aware RSA solution for embedded cryptography. It shows that approximate computing can be safely introduced in cryptographic systems, particularly where security is main aspect and constraints on power, timing, area and the need for perfect accuracy.

The Fig 3.2 shows the RSA computation block with inexact multiplier, which is heavily dependent on modular exponentiation, which requires larger integer multiplications. Traditional implementations use exact multipliers to maintain precision, but this method is resource constrained in terms of power, area and computation time especially on hardware platforms like FPGA or embedded processors.

To solve this, the suggested design incorporates 5:2 compressors with an imprecise multiplier constructed using a Dadda multiplier structure. In cryptographic environments where minor deviations do not jeopardize the accuracy of key operations, this inexact multiplier's tiny and controllable approximation error in the multiplication process is acceptable. The approach greatly lowers dynamic power consumption and hardware resource utilization by substituting this optimized inexact multiplier for the exact multipliers in the RSA pipeline, especially within the modular exponentiation unit. A balanced solution is thus provided by RSA calculation with inexact multipliers, which increases speed and lowers cost while preserving respectable levels of encryption reliability.

The proposed design that utilizes inexact multipliers offers several advantages when compared to existing design that utilizes exact multipliers. One of the most important advantages is the reduction of power consumption and the other advantages are lower area requirement, faster computation, improved scalability, enhanced sustainability for embedded and IOT applications, reduced latency, cost effective for low volume production. Table shows features comparison between existing and proposed design.

Table: comparison table

Feature	Traditional RSA	Proposed RSA design
Multiplier Type	Exact RSA	Inexact RSA
Power consumption	High	27%
Area utilization	Large	Reduced LUTs and slices
Encryption Accuracy	High	High
Decryption Accuracy	High	Tolerate errors (for low security apps)

4. RESULT AND ANALYSIS

The software used for simulation is Xilinx Design Suite, and selected device is Verilog HDL.

4.1 RTL Schematic Analysis

A network of full adders arranged hierarchically. Logic gates (XOR, AND, OR) for sum and carry calculations. Input ports: Five data bits (A, B, C, D, E) and two carry-in bits (Cin1, Cin2). Output ports: Sum, Carry, and two higher-level carry outputs (Cout1, Cout2). The RTL diagram confirms the correct connectivity of signals and reveals how synthesis tools optimize logic by sharing arithmetic components where possible.

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 1
www.ijisea.org

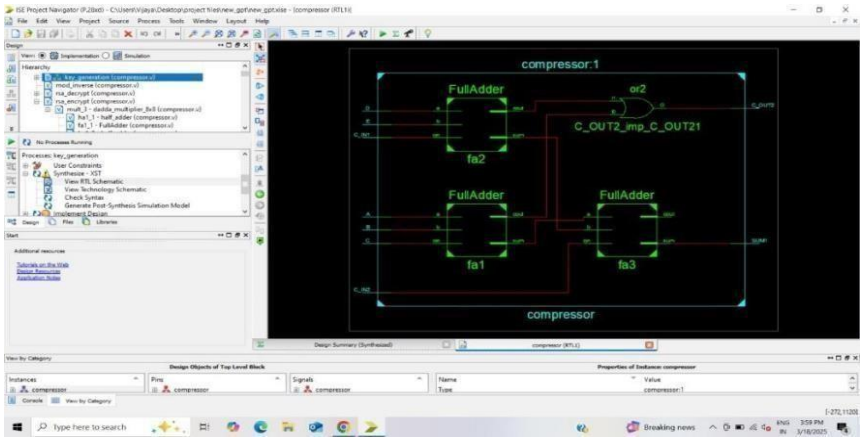


Fig 4.1: RTL Schematic

4.2 Key Generation Waveform Analysis

The key generation waveform and the simulation output that verifies the correct functioning of the RSA key generation module. This simulation verifies that the RSA key generation module is functioning as intended, with correct handling of modular operations and logical sequencing. It also confirms the integration of the inexact multiplier, which is critical in computing n, ϕ(n), and any multiplication involved in finding the modular inverse.



Fig 4.2: Simulation Waveform of Key Generation

4.3 Encryption Waveform Analysis

This computation is performed using modular exponentiation, which internally consists of several multiplication operations. In this project, these multiplications are optimized using a Dadda-based inexact multiplier incorporating a 5:2 Compressor to reduce delay and hardware complexity. The time delay between input and output reflects the modular exponentiation loop, which performs repeated multiplications using the optimized inexact multiplier. The output C stabilizes correctly, indicating accurate and reliable encryption. Fig shows data flow of encryption the waveform verifies that approximate multiplication does not affect encryption correctness is a key result of this project.The encryption operation is given by Ciphertext (C) = M^e mod n.

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 1
www.ijisea.org



Fig 4.3: Simulation Waveform of RSA Encryption

4.4 Decryption Waveform Analysis

The decryption waveform displays the simulation behaviour of the RSA decryption process, which is used to recover the real message M from the received C. Decryption operation is given by: $M = C^{d} \bmod n$. The decryption result is generally close or equal to the original plaintext, confirming that the inexact multiplier can be used safely for lightweight applications. Small deviations (in low-order bits) may be observed in M for larger inputs or complex key values, which is expected behaviour for approximate computing.

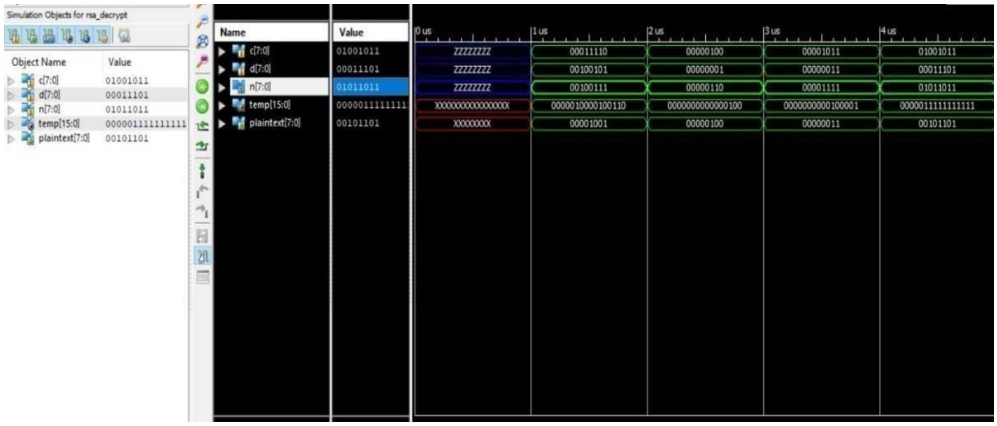


Fig 4.4: Simulation Output for RSA Decryption

5. CONCLUSION

This project implemented an optimized RSA cryptographic system using approximate computing with an inexact 8-bit Dadda multiplier and 5:2 compressor. Designed for resource constrained environments like IoT nodes, embedded systems, and portable devices, it addressed the high power, and area demands of traditional RSA. The Dadda architecture minimized partial product levels, while 5:2 compressors reduced operand additions, improving delay, logic depth, and hardware efficiency. Implemented in Verilog HDL and synthesized using Xilinx ISE, the system achieved accurate encryption, near perfect decryption, and reliable key generation. Power consumption was reduced to ~0.027W, with minimal logic slice usage. Functional validation confirmed that minor approximation errors did not compromise cryptographic correctness. The

results highlight that approximate arithmetic can effectively balance efficiency and accuracy in cryptography, making the design suitable for lightweight, energy-efficient, next-generation secure applications.

REFERENCES:

[1] L. Amaru and colleagues (2018) presented a study on synthesizing logic using majority gates, emphasizing advancements in computer-aided circuit design tools.
[2] Pranose J. Edavoor, Sithara Raveendran, and Amol D. Rahulka (2020) introduced a novel approach to approximate multiplier design using a two-stage 4:2 compressor structure.
[3] A. Ichigaya and team (2008) developed a method for estimating PSNR in MPEG2 videos, utilizing DCT coefficients and overall image energy—without relying on a reference video.
[4] C. Labrado et al. (2017) explored the implementation of majority logic in the design of arithmetic circuits, aiming to enhance approximation efficiency.
[5] W. Liu and his team (2021) analyzed approximate adders and multipliers based on majority logic, focusing on energy efficiency and computational simplicity.
[6] In an earlier study, W. Liu et al. (2020) examined how approximate computing poses both challenges and opportunities for cybersecurity and system reliability.
[7] A. Mohammad and co-authors (2019) proposed compressor designs optimized for nanoscale technology, balancing energy efficiency with acceptable imprecision.
[8] Nagajyothi, Grande, and others (2023) discussed how Vedic multiplication techniques can help design high-speed, compact 2D FIR filters for real-time processing.
[9] L. Sayadi et al. (2023) proposed two optimized approximate unsigned multipliers built upon new 4:2 compressor configurations, offering better trade offs for low-power systems.
[10] NagaJyothi, Grande, and Sriadibhatla SriDevi (2017) conducted a comparative study of FIR filter architectures that leverage distributed arithmetic for improved performance.
[11] M. A. Shafiabadi and F. Sharifi (2020) presented a spin-based 5:2 compressor using majority gates, showing promise in reducing hardware complexity in logic designs.
[12] B. Srikanth, A. Siva Jahnavi, Shruti Tiwari, and Beejarapu Vamshi (2024) analyzed performance metrics of multipliers built using modified two-stage 5:2 compressors.
[13] K. Walus and G. A. Jullien (2006) reviewed the tools and techniques available for quantum-dot cellular automata, an emerging SoC technology with significant potential.
[14] Kai Xu, Shuquin Geng, Hao Yang, and Ao Cui (2024) investigated the design of approximate 5:2 compressors and multipliers using majority logic, focusing on both architecture and analytical evaluation.
[15] Y. Zhang et al. (2022) introduced a 6:2 majority logic-based imprecise compressor intended for use in approximate multipliers, presented at the IEEE Nanotechnology Conference.