

वैदिक गणित द्वारा क्रिप्टोग्राफिक प्रगति: RSA प्रदर्शन विश्लेषण Cryptographic Advancements using Vedic Mathematics: RSA Performance Analysis

¹वैष्णवी भारद्वाज एवं ²आर्यमन भारद्वाज

¹Vaishnavi Bhardwaj and ²Aryaman Bhardwaj

¹Department of CSE, J.C Bose University of Science and Technology,
Faridabad, India

²Department of CSE, Manav Rachna International Institute of Research and Studies,
Faridabad, India

¹vbhardwaj1002@gmail.com, ²bhardwaj.aryaman.2005@gmail.com

<https://doi.org/10.5281/zenodo.21817668>

सारांश

RSA क्रिप्टोग्राफिक एल्गोरिद्म की गणनात्मक जटिलता, विशेष रूप से बड़े कुंजी आकारों (Key Sizes) के लिए, आधुनिक सुरक्षित संचार प्रणालियों में महत्वपूर्ण चुनौतियाँ प्रस्तुत करती है। इस समस्या के समाधान हेतु, हम पारंपरिक RSA एल्गोरिद्म और उसके एक अनुकूलित संस्करण का तुलनात्मक विश्लेषण प्रस्तुत करते हैं, जिसमें मॉड्यूलर गुणा (Modular Multiplication) के लिए वैदिक गणित की “ऊर्ध्व-तिर्यकभ्याम्” विधि का प्रयोग किया गया है। यह अध्ययन 1024, 2048, 4096 और 8192 बिट्स के विभिन्न कुंजी आकारों पर दोनों दृष्टिकोणों के प्रदर्शन का मूल्यांकन करता है, जिसमें मुख्य रूप से कुंजी निर्माण (Key Generation), एन्क्रिप्शन (Encryption) और डिक्रिप्शन (Decryption) प्रक्रियाओं पर ध्यान केंद्रित किया गया है। बड़े कुंजी आकारों पर, वैदिक-आधारित उन्नत RSA एल्गोरिद्म पारंपरिक RSA की तुलना में लगभग 56% अधिक तेजी से कुंजी उत्पन्न करता है, जिससे यह सीमित प्रसंस्करण शक्ति वाले वातावरण के लिए अत्यंत उपयुक्त सिद्ध होता है। यद्यपि एन्क्रिप्शन और डिक्रिप्शन समय दोनों विधियों में लगभग समान रहता है, किन्तु जैसे-जैसे कुंजी का आकार बढ़ता है, वैदिक विधि की गति और विस्तार क्षमता स्पष्ट रूप से दिखाई देती है। प्रयोगात्मक निष्कर्षों को विस्तृत तालिकाओं और दृश्य आरेखों के माध्यम से प्रस्तुत किया गया है, जो क्रिप्टोग्राफिक अनुकूलन में वैदिक गणित के व्यावहारिक लाभों को रेखांकित करते हैं। यह शोधपत्र वैदिक गणितीय सिद्धांतों के उपयोग से आधुनिक क्रिप्टोग्राफिक प्रणालियों को मजबूत बनाना दर्शाता है तथा सुरक्षा से समझौता किए बिना गणनात्मक प्रदर्शन को बेहतर बनाने का एक वैकल्पिक दृष्टिकोण प्रस्तुत करता है। पारंपरिक तकनीकों और समकालीन क्रिप्टोग्राफिक आवश्यकताओं के बीच सेतु निर्माण करते हुए, यह अध्ययन प्रभावी क्रिप्टोग्राफिक एल्गोरिद्मों के विकास में आगे के अनुसंधान और नवाचार की नींव रखता है।

Abstract

The computational complexity of the RSA cryptographic algorithm, particularly for large key sizes, poses challenges in modern secure communication systems. To address this, we present a comparative analysis of the traditional RSA algorithm and an optimized variant that employs the Urdhva Tiryakbham method from Vedic mathematics for modular multiplication. The study evaluates the performance of both approaches across varying key sizes (1024, 2048, 4096, and 8192 bits), focusing on key generation, encryption, and decryption processes. For higher key sizes, the enhanced

Vedic-based RSA generates keys up to 56% faster, which makes it ideal for settings with limited processing power. While encryption and decryption times remain comparable between the two methods, the scalability and efficiency of the Vedic-based approach become evident as the key size increases. Experimental findings, presented through detailed tables and visual charts, underscore the practical advantages of integrating Vedic mathematics into cryptographic optimization. This paper highlights the potential of ancient mathematical principles to enhance modern cryptographic systems, offering an alternative approach to improve computational performance without compromising security. By bridging traditional techniques and contemporary cryptographic demands, this study lays the groundwork for further investigation and invention in the development of effective cryptographic algorithms.

मुख्यशब्द: RSA क्रिप्टोग्राफी, वैदिक गणित अनुकूलन, ऊर्ध्व-तिर्यग्भ्याम् एल्गोरिद्म, सार्वजनिक कुंजी क्रिप्टोग्राफी दक्षता, क्रिप्टोग्राफी में मॉड्यूलर अंकगणित।

Key words: RSA Cryptography, Vedic Mathematics Optimization, Urdhva-Tiryagbhyam Algorithm, Public Key Cryptography Efficiency, Modular Arithmetic in Cryptography.

परिचय

सन् 1977 में रॉन रिबेस्ट, अदी शमीर और लियोनार्ड ऐडलमैन (RSA) ने RSA एल्गोरिद्म प्रस्तुत किया, जो एक क्रांतिकारी सार्वजनिक-कुंजी क्रिप्टोसिस्टम था और आधुनिक क्रिप्टोग्राफी की नींव रखने वाला सिद्ध हुआ [1]। यह विषमक (Asymmetric) एन्क्रिप्शन का पहला वास्तविक अनुप्रयोगों में से एक था। RSA दो कुंजियों के जोड़े का प्रयोग करता है: एन्क्रिप्शन के लिए सार्वजनिक कुंजी तथा डिक्రిप्शन के लिए निजी कुंजी। इसकी सुरक्षा का आधार बड़े संयोजित संख्याओं (Composite Numbers) के गुणनखंडन की गणनात्मक जटिलता है—एक ऐसी समस्या जिसे वर्तमान पारंपरिक एल्गोरिद्मों के माध्यम से पर्याप्त बड़े संख्याओं के लिए हल करना व्यावहारिक रूप से असंभव माना जाता है। इसी विशेषता के कारण RSA का प्रयोग आज कुंजी विनिमय (Key Exchange), डेटा एन्क्रिप्शन और डिजिटल हस्ताक्षरों जैसे क्षेत्रों में व्यापक रूप से किया जाता है।

एल्गोरिद्म मुख्य रूप से मॉड्यूलर अंकगणित (Modular Arithmetic) और अभाज्य संख्याओं के गुणनखंडन (Prime Factorization) पर आधारित है। एन्क्रिप्शन और डिक्రిप्शन की प्रक्रिया मॉड्यूलर घातांक (Modular Exponentiation) द्वारा संभव होती है,

जिससे प्रेषक और प्राप्तकर्ता दोनों को पहले से कोई गुप्त कुंजी साझा करने की आवश्यकता नहीं रहती। गतिशीलता में, RSA कई क्रिप्टोग्राफिक प्रोटोकॉल और सिस्टम का मुख्य आधार बन चुका है।

हालाँकि, जैसे-जैसे गणनात्मक शक्ति (Computational Power) बढ़ी है, RSA के प्रदर्शन-विशेषकर कुंजी निर्माण तथा एन्क्रिप्शन/डिक्రిप्शन की गति—पर प्रश्नचिह्न उठने लगे हैं। इस गणनात्मक भार को कम करने के लिए, विशेष रूप से मॉड्यूलर गुणा (Modular Multiplication) के क्षेत्र में, विभिन्न अनुकूलन (Optimizations) प्रस्तावित किए गए हैं [2]। इसके बावजूद, बड़े कुंजी आकारों के लिए RSA अब भी अत्यधिक गणनात्मक रूप से महंगा है, जो उच्च-थ्रूपुट वाले वातावरण में इसकी दक्षता को प्रभावित कर सकता है।

इसके अतिरिक्त, क्वांटम कंप्यूटिंग की बढ़ती संभावनाएँ RSA की सुरक्षा के लिए गंभीर खतरा प्रस्तुत करती हैं, क्योंकि शॉर का एल्गोरिद्म (Shor's Algorithm) बड़े पूर्णांकों का गुणनखंडन बहुपद समय (Polynomial Time) में कर सकता है [3]। ये चुनौतियाँ इस तथ्य को रेखांकित करती हैं कि RSA की प्रासंगिकता और प्रदर्शन को बनाए रखने के लिए निरंतर क्रिप्टोग्राफिक अनुकूलनों और वैकल्पिक विधियों की खोज आवश्यक है।

RSA एन्क्रिप्शन/डिक्रिप्शन में मॉड्यूलर गुणा का महत्व

RSA एल्गोरिदम मॉड्यूलर गुणा पर अत्यधिक निर्भर करता है, जो एन्क्रिप्शन और डिक्रिप्शन दोनों प्रक्रियाओं के लिए आवश्यक है।

RSA एन्क्रिप्शन मूल रूप से मॉड्यूलर घातांक का एक रूप है, जिसमें संदेश M को निम्न समीकरण द्वारा गुप्त संदेश C में परिवर्तित किया जाता है:

$C = m^e \bmod n$, जहाँ e सार्वजनिक घातांक (Public Exponent) है और n मॉड्यूलस है (दो बड़ी अभाज्य संख्याओं का गुणनफल)। इसी प्रकार, डिक्रिप्शन के दौरान, गुप्त संदेश C को निजी कुंजी घातांक d का उपयोग करके पुनः मूल संदेश M में परिवर्तित किया जाता है:

$$m = C^d \bmod n \quad [1].$$

RSA की दक्षता मुख्य रूप से मॉड्यूलर गुणा की गति पर निर्भर करती है, विशेषकर बड़े कुंजी आकारों के लिए। मॉड्यूलर घातांक प्रायः अनेक मॉड्यूलर गुणाओं को सम्मिलित करता है, और जैसे-जैसे मॉड्यूलस n का आकार बढ़ता है, ये क्रियाएँ गणनात्मक रूप से अत्यधिक कठिन होती जाती हैं। फलस्वरूप, मॉड्यूलर गुणा की गणनात्मक दक्षता में सुधार RSA के प्रमुख अनुकूलन क्षेत्रों में से एक है, विशेषकर उच्च-सुरक्षा अनुप्रयोगों (जैसे 2048 बिट या उससे अधिक कुंजी आकार) में। मॉड्यूलर गुणा को तेज करने के लिए विभिन्न विधियाँ प्रस्तावित की गई हैं, जैसे मॉन्टगोमेरी गुणन तकनीक (Montgomery Multiplication Technique), जो मॉड्यूलर रिडक्शन से संबंधित गणनात्मक लागत को कम करती है [4]। इन अनुकूलनों के बावजूद, मॉड्यूलर गुणा का अत्यधिक गणनात्मक भार अभी भी एक बॉटलनेक बना हुआ है, विशेषकर उन अनुप्रयोगों में जहाँ वास्तविक समय (Real-Time) या उच्च गति वाली क्रिप्टोग्राफिक क्रियाओं की आवश्यकता होती है।

पारंपरिक विधियों में गणनात्मक अधिभार की चुनौतियाँ

पारंपरिक RSA एल्गोरिदम के सामने सबसे बड़ी बाधा बड़े कुंजी आकारों की उच्च गणनात्मक लागत है, विशेषकर मॉड्यूलर घातांक की स्थिति में। एल्गोरिदम का बड़े पूर्णांकों का गुणन तथा मॉड्यूलर रिडक्शन पर अत्यधिक निर्भर होना, जैसे-जैसे कुंजी आकार बढ़ता है, उतना ही अक्षम हो जाता है। इसका परिणाम अत्यधिक प्रोसेसिंग समय और स्मृति उपयोग के रूप में सामने आता है, जो संसाधन-सीमित वातावरण—जैसे एम्बेडेड सिस्टम या सीमित गणनात्मक शक्ति वाले उपकरणों—के लिए अत्यधिक कठिनाई उत्पन्न करता है [2]।

इन चुनौतियों को आंशिक रूप से कम करने के लिए विभिन्न अनुकूलन तकनीकें विकसित की गई हैं, जैसे Exponentiation by Squaring और चीनी शेषफल प्रमेय (Chinese Remainder Theorem, CRT), जो आवश्यक गुणों की संख्या कम करके मॉड्यूलर घातांक की गति में सुधार कर सकती हैं [5]। फिर भी, ये सुधार मूल रूप से मॉड्यूलर संक्रियाओं की जटिलता द्वारा सीमित रहते हैं। उच्च स्तर की सुरक्षा की बढ़ती मांग—जैसे हमलों के विरुद्ध अधिक प्रतिरोधक क्षमता हेतु बड़े कुंजी आकार—इस समस्या को और गंभीर बना देती है, और इस बढ़ते गणनात्मक बोझ को सँभालने के लिए और अधिक कुशल विधियों की आवश्यकता उत्पन्न होती है।

इसके अतिरिक्त, जैसे-जैसे RSA की गणनात्मक आवश्यकताएँ बढ़ती हैं, वैसे-वैसे उसकी ऊर्जा खपत और विलंब भी बढ़ती है, विशेषकर वास्तविक समय (Real-Time) संचार परिदृश्यों में। हाल के वर्षों में इन समस्याओं को कम करने के लिए हार्डवेयर त्वरक और समानांतर प्रसंस्करण दृष्टिकोणों का अन्वेषण किया गया है, किंतु ये समाधान अपनी जटिलताओं के साथ आते हैं—जैसे हार्डवेयर लागत में वृद्धि और बड़े मॉड्यूलर संक्रियाओं को समानांतर करने की जटिलता [6]।

वैदिक गणित

वैदिक गणित, जो भारत की एक प्राचीन गणितीय तकनीकी प्रणाली है, ऐसे सिद्धांतों पर आधारित है जो जटिल अंकगणितीय क्रियाओं के लिए कुशल समस्या-समाधान विधियाँ प्रदान करती हैं। प्रक्रिया को छोटे और सरल घटकों में विभाजित करके, ऊर्ध्व-तिर्यक्याम् नामक गुणन सूत्र गुणा के लिए एक प्रभावी विधि प्रस्तुत करता है [7]। यह सूत्र आवश्यक गणनात्मक क्रियाओं की संख्या को कम करता है, जिससे विशेषकर बड़े अंकों के साथ कार्य करते समय, गुणा अधिक तीव्र गति से किया जा सकता है। इस विधि ने गणनात्मक अनुकूलन में महत्वपूर्ण लाभ सिद्ध किए हैं, विशेष रूप से डिजिटल सिग्नल प्रोसेसिंग और हार्डवेयर डिजाइन जैसे क्षेत्रों में [8]।

गणनात्मक अनुकूलन के संदर्भ में, वैदिक विधियाँ—जैसे ऊर्ध्व-तिर्यक्याम् का उपयोग तेज पूर्णांक गुणन और मॉड्यूलर अंकगणित के लिए किया गया है, जो RSA जैसे क्रिप्टोग्राफिक एल्गोरिद्मों के लिए केंद्रीय महत्व रखते हैं। ये विधियाँ आवश्यक क्रियाओं की संख्या को कम करने और गणनात्मक दक्षता में सुधार करने का एक माध्यम प्रदान करती हैं, विशेषकर उन परिस्थितियों में जहाँ बड़े अंकों के उच्च-श्रृंखला प्रसंस्करण की आवश्यकता होती है [9]।

हालाँकि, वैदिक तकनीकों का प्रयोग मैट्रिक्स गुणन और सिग्नल प्रोसेसिंग जैसे विशिष्ट क्षेत्रों में व्यापक रूप से किया गया है, परंतु क्रिप्टोग्राफिक अनुकूलन में उनका उपयोग अभी तक बड़े पैमाने पर अन्वेषित नहीं किया गया है।

कार्यप्रणाली

RSA एल्गोरिद्म का संक्षिप्त अवलोकन

RSA एल्गोरिद्म एक लोकप्रिय विषमक क्रिप्टोग्राफिक विधि है, जो दो कुंजियों का उपयोग करता है: निजी कुंजी – डिक्रिप्शन के लिए सार्वजनिक कुंजी – एन्क्रिप्शन के लिए इसकी प्रक्रिया मुख्यतः तीन प्रमुख चरणों में विभाजित होती है:

1. कुंजी निर्माण

2. एन्क्रिप्शन

3. डिक्रिप्शन

कुंजी निर्माण

1. सबसे पहले दो बड़े अभाज्य संख्याएँ p और q का चयन करें।

2. उनका गुणनफल ज्ञात करें:

$$n = p \times q$$

यह मान n सार्वजनिक और निजी दोनों कुंजियों का मॉड्यूलस होगा। एन्क्रिप्शन की शक्ति मुख्यतः n के परिमाण (Magnitude) पर निर्भर करती है।

3. ऑयलर का टोटिएंट फलन (Euler's Totient Function) की गणना करें:

$$\phi(n) = (p-1)(q-1)$$

4. एक एन्क्रिप्शन घातांक e चुनें, ताकि: $1 < e < \phi(n)$, और e तथा $\phi(n)$ परस्पर अभाज्य (Coprime) हों।

5. डिक्रिप्शन घातांक d की गणना करें, जो e का मॉड्यूलर व्युत्क्रम है $\phi(n)$ के मॉड्यूलो में।

दूसरे शब्दों में:

$$e \times d \equiv 1 \pmod{\phi(n)}$$

6. इस प्रकार, सार्वजनिक कुंजी (e, n) , निजी कुंजी (d, n) .

एन्क्रिप्शन:

जब दोनों कुंजियाँ (सार्वजनिक और निजी) उत्पन्न हो जाती हैं, तो RSA एन्क्रिप्शन की प्रक्रिया इस प्रकार होती है:

1. संदेश निरूपण : साधारण पाठ संदेश M को पहले एक पूर्णांक m के रूप में प्रस्तुत किया जाता है, जहाँ: $0 \leq m < n$

2. एन्क्रिप्शन: सार्वजनिक कुंजी (e, n) का उपयोग करके सांकेतिक पाठ (Ciphertext) C की गणना इस सूत्र द्वारा की जाती है:

$$C = m^e \bmod n$$

प्राप्त किया गया सांकेतिक पाठ C अब सुरक्षित रूप से संचारित (Transmit) किया जा सकता है।

डिक्रिप्शन :

निजी कुंजी (d, n) का उपयोग करके मूल संदेश को पुनः प्राप्त किया जाता है।

1. प्राप्तकर्ता अपने निजी कुंजी d का उपयोग करके सांकेतिक पाठ C से मूल संदेश m की गणना इस सूत्र द्वारा करता है:

$$m = C^d \bmod n$$

2. डिक्रिप्ट किया गया संदेश m पुनः अपने मूल साधारण पाठ रूप में परिवर्तित किया जाता है।

बड़े मिश्रित संख्याओं की गणना की चुनौती RSA एल्गोरिद्म की सुरक्षा का आधार है। (n, e) दिए जाने पर d की गणना करना असंभव है जब तक कि n को उसके अभाज्य गुणकों p और q में विभाजित न किया जाए।

बड़े कुंजी आकारों के लिए यह गुणनखंडन घातीय समय लेता है।

आरएसए के साथ ऊर्ध्व-तिर्यक्यम् का एकीकरण

वैदिक गणित की एक प्रमुख तकनीक ऊर्ध्व-तिर्यक्यम् सूत्र दो संख्याओं के कुशल गुणन हेतु प्रयुक्त होती है, जिसमें क्रॉसवाइज (आड़ा-तिरछा) गुणन लागू किया जाता है। इस विधि में आंशिक गुणनफलों की गणना समानांतर रूप से की जाती है और तत्पश्चात उन्हें जोड़कर अंतिम गुणनफल प्राप्त किया जाता है।

गुणन की आवश्यक चरणों की संख्या को घटाकर, यह विधि उन क्रिप्टोग्राफिक एल्गोरिद्म में संचालन की गति बढ़ा सकती है जिनमें बड़े संख्याओं का गुणन आवश्यक होता है, जैसे कि RSA एल्गोरिद्म।

ऊर्ध्व-तिर्यक्यम् सूत्र की व्याख्या:

ऊर्ध्व-तिर्यक्यम् सूत्र का मूल सिद्धांत निम्नलिखित चरणों पर आधारित है:

- आड़ा-तिरछा गुणन :** दो संख्याओं $a = a_1, a_2, \dots, a_n$ और $b = b_1, b_2, \dots, b_n$ के लिए, गुणन इस प्रकार किया जाता है कि दोनों संख्याओं के अंकों को जोड़ा जाता है और उनका गुणन किया जाता है।
- कैरीओवर प्रबंधन :** जैसे-जैसे आंशिक गुणनफल (Partial Products) की गणना होती है, कोई भी अधिशेष (Overflow) अगले उच्च स्थान मान (Higher Place Value) में जोड़ा जाता है।
- अंतिम परिणाम का निर्माण :** आंशिक गुणनफल की गणना कर उसे जोड़ने के बाद, परिणाम को पलटकर अंतिम गुणनफल प्राप्त किया जाता है।

इस विधि को `vedic_multiply(a, b)` फंक्शन में लागू किया गया है, जिसमें:

- इनपुट संख्याओं a और b को पहले पलटा जाता है।
- आड़ा-तिरछा गुणन लागू किया जाता है और कैरीओवर को संभाला जाता है।
- अंतिम गुणनफल प्राप्त करने के लिए परिणाम को पुनः पलटा जाता है।

एल्गोरिद्म 1 – ऊर्ध्व-तिर्यकभ्याम् (UT) बड़ी संख्या गुणन

```

Algorithm( $A[0..n-1], B[0..m-1], \text{BASIS}$ )

  let  $P[0..n+m-1] \leftarrow 0$ 
  carry  $\leftarrow 0$ 
  //परिणाम में अधिकतम  $n+m$  अंक हो सकते हैं; इसे कॉलम – दर – कॉलम गणना करें
  for  $k$  from 0 to  $n+m-2$  do
     $S \leftarrow \text{carry}$ 
    //कॉलम  $k$  के लिए आड़ा – तिरछा संचयन
     $i_{\min} \leftarrow \max(0, k-(m-1))$ 
     $i_{\max} \leftarrow \min(k, n-1)$ 
     $i_{\min}$  से  $i_{\max}$  तक करें:
       $j \leftarrow k-i$ 
       $S \leftarrow S + A[i] * B[j]$ 
  अंत
   $P[k] \leftarrow S \bmod \text{BASIS}$ 
  carry  $\leftarrow [S / \text{BASIS}]$ 
end for
 $P[n+m-1] \leftarrow \text{carry}$ 
return Normalize ( $P$ )/अग्रिम शून्य (Leading Zeros) हटाएँ
  
```

कार्यान्वयन नोट्स:

- यदि आप $\text{BASIS} = 2^w$ (e.g., $w=32$ या 64) चुनते हैं, तो यह मशीन वर्ड अंकगणित के साथ मेल खाता है और किसी भी आधार से विभाजन की महंगी गणनाओं से बचाता है।
- केवल दशमलव प्रस्तुति के लिए $B=10$; खें; उच्च प्रदर्शन कोड के लिए वर्ड-स्तरीय आधार $B=2^w$ का उपयोग करना बेहतर है।

एल्गोरिद्म 2 – UT-Multiply Mod M (मॉड्यूलर अंकगणित के लिए)

```

Algorithm UT-Multiply-Mod ( $A, B, M, \text{BASIS}$ )

  // आवश्यकतानुसार तेज रिडक्शन के लिए  $M'$  को पूर्व-गणना किया जा सकता है।
   $P \leftarrow 0$  // बड़ी संख्या संचितकर्ता
  carry  $\leftarrow 0$ 
  
```

```

k = 0 से len(A) + len(B) - 2 तक करें:
    S ← carry
    i_min ← max(0, k - (len(B) - 1))
    i_max ← min(k, len(A) - 1)
    for i from i_min to i_max do
        j ← k - i
        S ← S + A[i] * B[j]
    अंत
    // k अंक निकालें और कैरी अपडेट करें
    digit_k ← S mod BASIS
    carry ← [S / BASIS]
    // digit_k को बड़े अंक P में स्थिति k पर सम्मिलित करें
    P ← P + digit_k * (BASIS^k)
    if (k mod r == 0) then
        // उदाहरण के लिए, r= वर्ड आकार या छोटा स्थिरांक
        P ← P mod M
    अंत यदि
अंत
P ← P + carry * (BASIS^(len(A) + len(B) - 1))
return P mod M

```

RSA में संशोधन:

परंपरागत RSA में कुंजी निर्माण प्रक्रिया दो अभाज्य संख्याओं p और q के गुणन पर आधारित होती है, ताकि $N = p \times q$ और ऑयलर का टोटिएंट फलन $\phi(N) = (p-1)(q-1)$ की गणना की जा सके। ऊर्ध्व-तिर्यक्भ्याम् विधि के माध्यम से वैदिक गुणन को सम्मिलित करके, इन गुणनों का गणनात्मक भार कम किया जाता है, विशेष रूप से बड़ी संख्याओं के लिए। संशोधित RSA कुंजी निर्माण चरण इस प्रकार हैं:

1. **अभाज्य संख्या निर्माण:** p और q उत्पन्न करने के लिए मानक विधियों का उपयोग किया जाता है।
2. **वैदिक गुणन:** $N = p \times q$ और $((p-1) \times (q-1))$ की गणना `vedic_multiply(a, b)` फ़ंक्शन का उपयोग करके की जाती है। यह विधि विशेष रूप से बड़ी p और q के लिए प्रक्रिया को अधिक कुशल बनाती है।

3. **मॉड्यूलर अंकगणित:** RSA एल्गोरिद्म का शेष भाग अपरिवर्तित रहता है। सार्वजनिक घातांक e इस प्रकार चुना जाता है कि $\gcd(e, \phi(N)) = 1$, हो। निजी घातांक d e का $\phi(N)$ के मॉड्यूलो में व्युत्क्रम होता है।

परंपरागत गुणन को ऊर्ध्व-तिर्यक्भ्याम् आधारित गुणन से बदलकर, RSA एल्गोरिद्म की दक्षता विशेष रूप से बड़ी कुंजी आकारों के लिए बढ़ जाती है। यह विधि गणनात्मक भार को काफी कम करती है, जिससे एन्क्रिप्शन और डिक्रिप्शन प्रक्रियाएँ तेज होती हैं। यह आधुनिक क्रिप्टोग्राफिक अनुप्रयोगों में अत्यंत महत्वपूर्ण है, जहाँ बड़े डेटा वॉल्यूम को संभालना आवश्यक होता है।

एल्गोरिद्म 3 – UT गुणन के साथ RSA कुंजी निर्माण

Algorithm RSA – KeyGen – UT(k, BASIS)

```

repeat
    p ← Generate Probable Prime(k / 2)
    q ← Generate Probable Prime(k / 2), q ≠ p
until p and q pass strong probable – prime tests
// N = p*q via UT
N ← UT – Multiply(p, q, BASIS)
// phi = (p-1) * (q-1) via UT
p1 ← p-1
q1 ← q-1
phi ← UT – Multiply(p1, q1, BASIS)
//choose e
e ← 65537
if gcd(e, phi) ≠ 1 then
    e ← Find Small Public Exponent Coprime To(phi)
end if
// d = e ^ {-1} mod phi
d ← Mod Inverse(e, phi)
return ((e, N), (d, N))
    
```

एल्गोरिद्म 4 – RSA एन्क्रिप्शन (Square-an-Multiply UT के साथ)

Algorithm RSA-Encrypt-UT(M,e,N,BASIS)

$C \leftarrow 1$

$\text{base} \leftarrow M \bmod N$

$\text{exp} \leftarrow e$

while $\text{exp} > 0$ do

 if $(\text{exp} \& 1) = 1$ then

$C \leftarrow \text{UT-Multiply-Mod}(C, \text{base}, N, \text{BASIS})$

 end if

$\text{base} \leftarrow \text{UT-Multiply}$

$-\text{Mod}(\text{base}, \text{base}, N, \text{BASIS})$

$\text{exp} \leftarrow \text{exp} \gg 1$

end while

return C

एल्गोरिद्म 5 – RSA डिक्रिप्शन (CRT वैकल्पिक) UT के साथ

Algorithm RSA – Decrypt – UT(C,d,N,BASIS)

$M \leftarrow 1$

$\text{base} \leftarrow C \bmod N$

$\text{exp} \leftarrow d$

while $\text{exp} > 0$ do

 if $(\text{exp} \& 1) = 1$ then

$M \leftarrow \text{UT-Multiply-Mod}(M, \text{base}, N, \text{BASIS})$

 end if

$\text{base} \leftarrow \text{UT-Multiply}$

$-\text{Mod}(\text{base}, \text{base}, N, \text{BASIS})$

$\text{exp} \leftarrow \text{exp} \gg 1$

end while

return M

तुलना के लिए मेट्रिक्स

इस खंड में, हम उन पैरामीटर्स और मेट्रिक्स को परिभाषित करते हैं जो पारंपरिक RSA और वैदिक RSA एल्गोरिद्म दोनों के प्रदर्शन का मूल्यांकन करने के लिए उपयोग किए जाते हैं। तुलना के लिए निम्नलिखित मुख्य मेट्रिक्स का उपयोग किया गया है:

1. **कुंजी निर्माण समय** : यह मापता है कि सार्वजनिक और निजी कुंजी बनाने में कितना समय लगता है। RSA-आधारित एन्क्रिप्शन योजनाओं में यह एक महत्वपूर्ण चरण है। रीयल-टाइम प्रदर्शन वाली प्रणालियों के लिए तेज कुंजी निर्माण समय वांछनीय है। इसे सेकंड में मापा जाता है, और कम समय अधिक कुशल कुंजी निर्माण को दर्शाता है।
2. **एन्क्रिप्शन समय** : यह मेट्रिक उस समय को कैप्चर करता है जो डेटा को सार्वजनिक कुंजी का उपयोग करके एन्क्रिप्ट करने में लगता है। चूंकि एन्क्रिप्शन गणनात्मक रूप से भारी हो सकता है, इसलिए तेज एन्क्रिप्शन समय बेहतर सिस्टम प्रदर्शन प्रदान करता है, विशेष रूप से बड़े डेटा सेट वाले अनुप्रयोगों

के लिए।

3. **डिक्रिप्शन समय** : यह वह समय है जो डेटा को निजी कुंजी का उपयोग करके डिक्रिप्ट करने में लगता है। एन्क्रिप्शन की तरह, डिक्रिप्शन भी यह सुनिश्चित करने के लिए महत्वपूर्ण है कि संवेदनशील डेटा सुरक्षित और कुशलतापूर्वक पहुँचा जा सके। कम डिक्रिप्शन समय सिस्टम की प्रतिक्रिया क्षमता को बेहतर बनाता है।
4. **डिक्रिप्शन सफलता** : यह सरल रूप से यह इंगित करता है कि डिक्रिप्ट किया गया डेटा मूल प्लेनटेक्स्ट से मेल खाता है या नहीं। दोनों RSA एल्गोरिद्म से उम्मीद की जाती है कि वे सही और वैध आउटपुट प्रदान करें।

RSA का मूल्यांकन विभिन्न कुंजी आकारों (1024, 2048, 4096, और 8192 बिट्स) के साथ किया गया। यह तुलना यह दर्शाने में मदद करती है कि प्रत्येक एल्गोरिद्म विभिन्न एन्क्रिप्शन जटिलताओं के तहत कैसा प्रदर्शन करता है। इन पैरामीटर्स में कुंजी निर्माण, एन्क्रिप्शन और डिक्रिप्शन समय शामिल हैं, जिन्हें प्रत्येक मामले में गणना करके पारंपरिक और वैदिक RSA

विधियों की गणनात्मक दक्षता का आकलन किया जाता है।

अगले खंड में, हम इन मेट्रिक्स के लिए प्राप्त परिणामों का विश्लेषण करेंगे।

परिणाम विश्लेषण

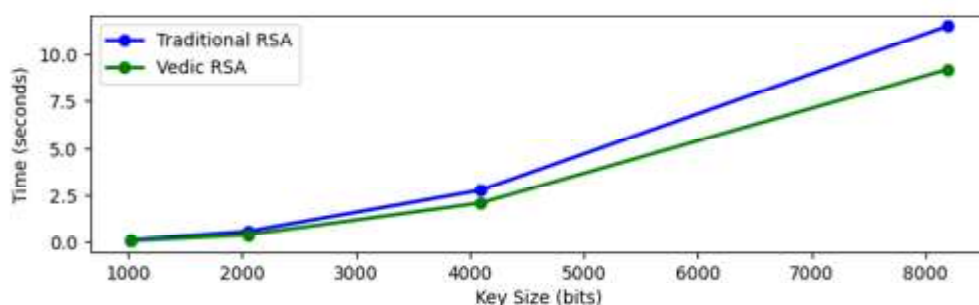
इस अध्ययन में, हमने पारंपरिक RSA और वैदिक RSA की प्रभावशीलता का मूल्यांकन किया, जिसमें प्रमुख मेट्रिक्स जैसे कुंजी निर्माण, एन्क्रिप्शन और डिक्रिप्शन के निष्पादन समय का उपयोग किया गया।

परिणाम नीचे दी गई तालिका में प्रस्तुत किए गए हैं, जो विभिन्न कुंजी आकारों (1024, 2048, 4096 और 8192 बिट्स) के साथ कार्य करते समय इन दोनों विधियों के बीच अंतर को उजागर करते हैं।

तालिका 1

कुंजी निर्माण प्रदर्शन

कुंजी आकार	पारंपरिक RSA समय (s)	वैदिक RSA समय (s)	समय अंतर/सुधार (%)
1024	0.0410	0.0589	43.44%
2048	0.3845	0.5366	39.53%
4096	7.9894	4.2354	-47.00%
8192	73.4741	32.0741	-56.35%



चित्र. 1. कुंजी निर्माण प्रदर्शन का ग्राफिकल प्रतिनिधित्व

कुंजी निर्माण:

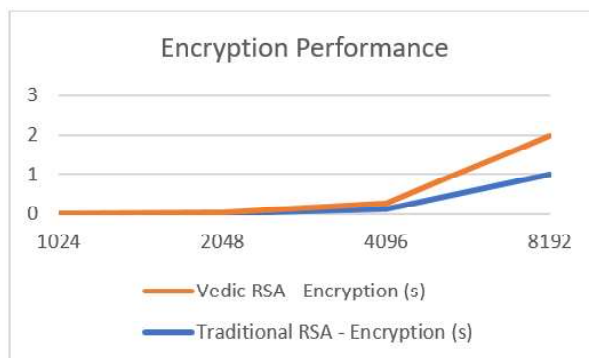
- तालिका 1 में, 1024-बिट कुंजी के लिए, वैदिक RSA ने पारंपरिक RSA की तुलना में 43.44% अधिक समय लिया, जबकि 2048-बिट कुंजी के लिए समय का अंतर 39.53% था।
- हालांकि, बड़े कुंजी आकारों (4096 और 8192 बिट्स) के लिए, वैदिक RSA ने कुंजी निर्माण समय में सुधार दिखाया। 4096-बिट कुंजी के लिए यह 47% तेज प्रदर्शन प्रदान करता

है, और 8192-बिट कुंजी के लिए 56.35% सुधार दिखाया। छोटे संख्याओं (1024 और 2048-बिट ऑपरेण्ड्स) के लिए, सेटअप ओवरहेड (जैसे विभाजन, पुनरावर्ती कॉल, पुनः संयोजन, अतिरिक्त जोड़) फाई गुणा में बचाए गए समय से अधिक होता है। यह दर्शाता है कि वैदिक RSA बड़े कुंजी आकारों के लिए अधिक उपयुक्त है, संभवतः इसकी गुणन तकनीक की उच्च गणनात्मक भार पर दक्षता के कारण।

तालिका 2

एन्क्रिप्शन प्रदर्शन

पारंपरिक RSA समय (s)	(पारंपरिक RSA समय) (s)	समय अंतर (%)
0.0030	0.0034	13.33%
0.0270	0.0235	-12.96%
0.1439	0.1200	-13.63%
0.9994	0.8295	-17.00%

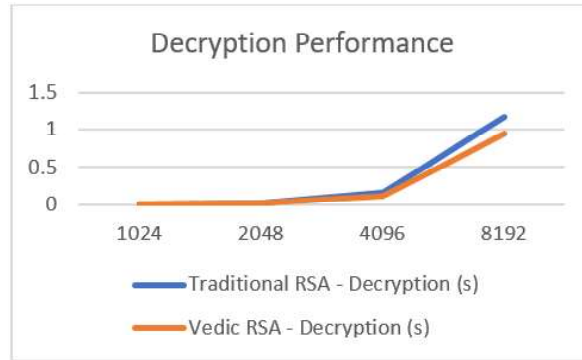
**एन्क्रिप्शन प्रदर्शन**

एन्क्रिप्शन:

- जैसा कि तालिका 2 में दिखाया गया है, 1024-बिट कुंजी के लिए, वैदिक RSA पारंपरिक RSA की तुलना में 13.33% धीमा था। यह छोटे कुंजी आकारों के लिए वैदिक गुणन विधि की जटिलता को दर्शाता है।
- 2048-बिट कुंजी पर, वैदिक RSA ने पारंपरिक RSA की तुलना में 12.96% तेज एन्क्रिप्शन समय प्रदर्शित किया, जो मध्यम आकार की कुंजी के लिए एक स्पष्ट लाभ को दर्शाता है।

तालिका 3**डिक्रिप्शन प्रदर्शन**

पारंपरिक RSA समय (s)	वैदिक RSA समय (s)	समय अंतर (%)
0.0042	0.0041	-2.44%
0.0200	0.0155	-22.50%
0.1581	0.1205	-23.74%
1.1670	0.8873	-23.97%

**डिक्रिप्शन:**

जैसा कि तालिका 3 में दिखाया गया है, डिक्रिप्शन प्रदर्शन के मामले में, वैदिक RSA लगातार पारंपरिक RSA से बेहतर प्रदर्शन करता है, और सभी परीक्षण किए गए कुंजी आकारों में सुधार दिखाता है। 1024-बिट कुंजी पर, वैदिक RSA 2.44% तेज है। जैसे-जैसे कुंजी का आकार बढ़ता है, डिक्रिप्शन समय में बचत और भी अधिक महत्वपूर्ण हो जाती है। 2048-बिट कुंजी के लिए सुधार: 22.50%, 4096-बिट कुंजी के लिए सुधार: 23.74%, 8192-बिट कुंजी के लिए सुधार: 23.97%

ये परिणाम वैदिक RSA दृष्टिकोण की क्षमता को दर्शाते हैं कि यह डिक्रिप्शन प्रक्रियाओं को तेज कर सकता है, संभवतः मॉड्यूलर घातांक चरण में बड़े संख्याओं

को कुशलतापूर्वक संभालने की वजह से। विशेष रूप से बड़े कुंजी आकारों के लिए यह डिक्रिप्शन समय में कमी महत्वपूर्ण है, जहाँ पारंपरिक RSA आमतौर पर बढ़ती गणनात्मक जटिलता से प्रभावित होता है।

ये निष्कर्ष संकेत देते हैं कि जबकि वैदिक RSA हमेशा छोटे कुंजी आकारों पर पारंपरिक RSA से बेहतर नहीं हो सकता—विशेषकर कुंजी निर्माण और एन्क्रिप्शन के लिए—फिर भी यह डिक्रिप्शन और स्केलेबिलिटी में महत्वपूर्ण लाभ प्रदान करता है।

इससे वैदिक RSA उन अनुप्रयोगों के लिए अधिक प्रभावी विकल्प बन सकता है, जहाँ डिक्रिप्शन की गति महत्वपूर्ण है और कुंजी आकार बढ़े हैं, जैसे कि सुरक्षित संचार प्रणालियाँ और ब्लॉकचेन पर्यावरण।

निष्कर्ष और भविष्य का दायरा

अध्ययन के निष्कर्ष दिखाते हैं कि पारंपरिक RSA तकनीक की तुलना में, वैदिक आधारित RSA गणनात्मक दक्षता और समय जटिलता में महत्वपूर्ण लाभ प्रदान करता है।

मुख्य बिंदु:

1. वैदिक गुणन तकनीक का उपयोग करके, विशेष रूप से बड़े अभाज्य संख्याओं के लिए, तेज गणना संभव होती है।
2. इससे मॉड्यूलर घातांक और गुणन से जुड़ा सामान्य ओवरहेड कम हो जाता है।
3. साथ ही, इस विधि से क्रिप्टोग्राफिक संचालन की सटीकता और शुद्धता बनी रहती है, जिससे संशोधित एल्गोरिदम सुरक्षित रहता है।

भविष्य का दायरा

हालांकि वर्तमान शोध वैदिक आधारित RSA की व्यवहार्यता को दर्शाता है, कुछ क्षेत्रों में और अनुसंधान आवश्यक है:

1. बड़े कुंजी आकारों के लिए अनुकूलन:

जैसे-जैसे RSA कुंजी का आकार बढ़ता है, गणनात्मक मांग भी बढ़ती है। भविष्य में वैदिक RSA को बड़े कुंजी आकारों के लिए और अधिक कुशल और स्केलेबल बनाने पर

काम किया जा सकता है इसमें उन्नत गुणन एल्गोरिदम का एकीकरण किया जा सकता है, ताकि समय जटिलता और गणनात्मक दक्षता के बीच संतुलन बना रहे।

2. वैदिक विधियों के लिए समानांतर प्रसंस्करण के अवसर:

उर्ध्व-तिर्यकभ्यां सूत्र की अंतर्निहित संरचना समानांतर प्रसंस्करण की संभावना सुझाती है। समानांतर तकनीकों की जांच करने से RSA संचालन और भी तेज हो सकते हैं, जिससे यह बड़े डेटा और वास्तविक समय अनुप्रयोगों के लिए अधिक उपयुक्त बनता है।

3. सुरक्षा प्रभाव और वास्तविक डेटा पर आगे का परीक्षण:

हालांकि वैदिक आधारित RSA एल्गोरिदम आशाजनक है, इसकी मजबूती विभिन्न हमले के तरीकों के खिलाफ सुनिश्चित करने के लिए व्यापक सुरक्षा विश्लेषण आवश्यक है। भविष्य के अध्ययन में वास्तविक दुनिया के डेटासेट्स पर कठोर परीक्षण शामिल होना चाहिए, ताकि एल्गोरिदम का प्रदर्शन और सुरक्षा व्यावहारिक परिदृश्यों में मूल्यांकित की जा सके।

Alphabetically sorted terminology in English	वर्णमाला अनुक्रमित हिंदी शब्दावली
Asymmetric Encryption	असममित गुप्तीकरण
Basis	आधार
Carry	वहन
Ciphertext	गुप्तलेख
Chinese Remainder Theorem (CRT)	चीनी शेषफल प्रमेय
Composite Numbers	संयुक्त संख्याएँ
Computational Efficiency	गणनात्मक दक्षता
Computational Overhead	गणनात्मक अधिभार
Computational Power	गणनात्मक शक्ति
Coprime	परस्पर अभाज्य
Cryptography	क्रिप्टोग्राफी

Decryption	गोप्य उद्घाटन
Digital Signatures	डिजिटल हस्ताक्षर
Encryption	गुप्तीकरण
Euler's Totient Function	आइलर टोटिएंट फलन
Exponentiation by Squaring	वर्गण द्वारा घातांकन
Exponential Time	घातीय समय
Factorization	गुणनखंडन
Fast Integer Multiplication	तीव्र पूर्णांक गुणन
Generate Probable Prime	संभाव्य अभाज्य उत्पन्न करना
GCD (Greatest Common Divisor)	महत्तम समापवर्तक
Hardware Acceleration	हार्डवेयर त्वरण
Key Exchange	कुंजी आदान-प्रदान
Key Generation	कुंजी उत्पत्ति
Latency	विलंब
Leading Zeros	अग्रग शून्य
Message Representation	संदेश निरूपण
Modular Arithmetic	प्रतिरूपक अंकगणित
Modular Exponentiation	प्रतिरूपक घातांकन
Modular Inverse	प्रतिरूपक व्युत्क्रम
Modular Multiplication	प्रतिरूपक गुणन
Modulus	अवशेषक
Montgomery Multiplication	मॉन्टगोमेरी गुणन
Overflow	अतिप्रवाह
Parallel Processing	समांतर प्रसंस्करण
Partial Products	आंशिक गुणनफल
Plaintext	मूललेख
Polynomial Time	बहुपद समय
Prime Factorization	अभाज्य गुणनखंडन
Private Key	निजी कुंजी
Public Key Cryptosystem	सार्वजनिक कुंजी कूटलेखन प्रणाली
Public Key	सार्वजनिक कुंजी
Public Exponent	सार्वजनिक घातांक
Real-Time	वास्तविक समय
Scalability	विस्तारिता
Shor's Algorithm	शोर का एल्गोरिद्म
Square-and-Multiply	वर्गण-गुणन
Urdhva Tiryakbhyam Sutra	ऊर्ध्व-तिर्यग्भ्याम सूत्र
Vedic Mathematics	वैदिक गणित

संदर्भ

1. Anu-Qareme K, ed. (1998) "School of Communication Studies, University of Ghana", Perspectives in Indigenous Communication in Africa; Volume 1, 58–60.
1. Rivest, R., Shamir, A. and Adleman, L. (1978) "A method for obtaining digital signatures and public-key cryptosystems," Communications of the ACM, vol. 21, no. 2, pp. 120-126, doi: 10.1145/359340.359342.
2. Green, M.B. (2012) "RSA public key cryptosystem and its performance improvements," International Journal of Computer Applications, vol. 45, no. 13, pp. 5-9, doi: 10.5120/6924-9635.
3. Shor, P. W. (1997) "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," SIAM Journal on Computing, vol. 26, no. 5, pp. 1484-1509, doi: 10.1137/S0097539795291114.
4. Pomerance, C. (2001) "The Montgomery multiplication method and its applications," Mathematics of Computation, vol. 70, no. 233, pp. 1523-1535, doi: 10.1090/S0025-5718-01-01346-0.
5. Dada, M.A.H. (2013) "Optimization techniques in modular exponentiation for RSA," Journal of Cryptographic Engineering, vol. 3, no. 2, pp. 101-109, doi: 10.1007/s13389-013-0053-2.
6. Varma, K.R. (2017) "Efficient RSA cryptosystem implementation using hardware acceleration," Journal of Computer Networks and Communications, vol. 2017, Article ID 6249471, pp. 1-8, doi: 10.1155/2017/6249471.
7. Sharma, J.S. (2005) "Vedic Mathematics: A Mathematical Method for the Modern World", 2nd ed., New Delhi: M. G. Publishers.
8. Jain, S.K. (2019) "Applications of Vedic mathematics in digital signal processing," Journal of Mathematical and Computational Sciences, vol. 12, no. 2, pp. 57-64, doi: 10.1080/0973152X.2019.1659243.
9. Gupta, M.S. (2018) "Efficient modular multiplication using Vedic mathematics," International Journal of Cryptography and Security, vol. 6, no. 4, pp. 230-239, doi: 10.1016/j.jcse.2018.01.005.
10. Kapoor, H.S. (2021) "Applications of Vedic Mathematics in Cryptography," International Journal of Computational Science, vol. 2, no. 1, pp. 34-47.
11. Sandhu, R.S. (2021) "Optimizing Cryptographic Algorithms Using Vedic Multiplication," Journal of Cryptographic Engineering, vol. 9, no. 2, pp. 93-101.
12. Pande, A.S. and Awasare, P.S. (2011) "A New Architecture for RSA Algorithm Using Vedic Mathematics," IEEE Xplore.
13. Rao, S.R. and Prasad, G.N.R. (2014) "An Efficient Hardware Model for RSA Encryption System Using Vedic Mathematics," ResearchGate.
14. Chouhan, A.S., Shekhar, G. and Mangal, A. (2025) "Solution of Various Programming Problems by Matrix Inverse Method", Vigyan Prakash, Vol. 23, Issue 2, pp 39-46.