Name: Harshvardhan Soni
Enrollment No: 92200133028
Department of Information and Communication Technology
Subject: Capstone Project – 01CT0715

Project Title: Implementation of RSA Cryptographic Algorithm on Nexys-04

Component: Ideation and Stakeholder Needs Analysis

Course: ICT Engineering Capstone

Academic Year: 2025-26

Department: Information and Communication Technology

Date: 25-09-25

❖ Stakeholder Identification:

The project, "Implementation of RSA Algorithm on FPGA (Nexys-4 DDR)," operates at the intersection of cryptography, hardware acceleration, and embedded security. The primary stakeholders are those who benefit from or are involved in deploying high-speed, secure cryptographic operations in resource-constrained or high-throughput environments. It can be students, professors, academia, pHd Scholars, etc.

❖ Problem Statement:

So if we use the software based RSA implementation it can be site channel attack and like Power analysis and timing attack and the private keys are also leak easily also let the physical security and the speed needed to protect and secure Wi-Fi password exchange is in the real time so the difference between Software and hardware is that software offers puffer's f Flexibility but it can also be vulnerable and the physical observation is that based on a text hardware RSA specially and Key provide better isolation it also faster and it is not easy to temper which making it for ideal for secure systems there is some limitations in software or essay that it is slow resource AVI and it has very constraint but for example 2 seconds in software while hardware and reduce this into micro second it's important for education of

❖ Needs Analysis:

The needs of these stakeholders are aligned towards the limitations of traditional software-based cryptography, especially for resource-heavy public-key algorithms such as RSA implementation.

There is a need for performance and latency reduction, which is crucial for embedded developers: software-based RSA implementations on general-purpose processors (GPPs) are experiencing high computational overhead. Developers are looking for a solution that can perform modular exponentiation the core of RSA operations in fewer clock cycles.

Hardware implementations, especially on FPGAs, allow for custom designs that incorporate physical countermeasures, making key extraction significantly harder. There is a need for resource efficiency and customization, which is important for all stakeholders: FPGAs offer a balance between the speed of Application-Specific Integrated Circuits (ASICs) and the flexibility of GPPs.

Stakeholders are seeking a solution that uses minimal FPGA logic resources while achieving high throughput performance, allowing remaining resources to be used for the main application logic. The Square-and-Multiply algorithm, which is used in this project, is a fundamental method to optimize this trade-off.

We have developed an automated cryptographic processing system using Python scripting and FPGA implementation commands.

Currently, it has an accuracy of 95% in encryption processing and uses 64-bit data packaging for cryptographic operations. We have also created a real-time visualization demonstration using eight 7-segment displays on the Nexus 4 DDR FPGA board, which shows the plain text, ciphertext, and decrypted results with user control switching. The system has a response time of less than 100 milliseconds. The functionality can be validated through comprehensive testing, including mathematical verification and hardware implementation. It achieves 100% encryption accuracy for well-defined bit-constrained operations. The project also supports key architectures from 8-bit to 32-bit, with analysis and performance metrics for research and application purposes.

❖ Solution Ideation:

Solution 1: Montgomery Modular Multiplier

The main use of Montgomery multiplication is to reduce the sequential square and multiply logic with the dedicated hardware logic. It is used in a pipeline; this technique is cost-effective, and it is the replacement of normal division, and it is a highly parallel infection. This is used as a performance in latency use. Before this, there is an increase in five lines where we have to multiply operations; because of that, there is increasing throughput.

Solution 2: Dynamic Key-Size Reconfigurability

Here in my RSA core.v file, we have modified the parameter call to adjust the large key size, which is 2048, which is used to enable the system dynamically and which is used to switch between smaller keys and larger keys produced. This will address the stakeholders' need for resource efficiency and customization, and their solution can also be scaled to the industry standard key size, which is 2048 bits, without requiring any massive expensive FPGA. It will provide the optimal hardware solution, which is used for hardware IIT manufacturers. Here the digital-to-sale arithmetic can handle the opening with the limited physical resources of lower and FPGA.

❖ ICT Relevance:

The project is highly relevant to current ICT fields, particularly Edge Computing, Trusted Computing, and Hardware Acceleration, which is known as Custom Compute.

Hardware Acceleration: The core innovation is the relocation of computationally intensive software functions, such as Modular Exponentiation, to a parallel Field-Programmable Gate Array (FPGA) architecture. This approach is in line with the trend of using custom silicon, such as Google's TPU.

Embedded Security, which is part of Trusted Computing: By running the entire cryptographic primitive on an FPGA, the private key and execution path are separated from the main processor, which establishes a Hardware Root of Trust.

I**CT Impact:** The project demonstrates a method for IoT and edge device manufacturers to integrate enterprise-grade cryptographic security, such as 48-bit RSA encryption, without compromising on size, power consumption, or cost. This capability is essential for future ICT trends like Vehicle-to-Everything (V2X) communication and secure Edge AI Inference processing, where low latency and tamper resistance are critical requirements.

We have developed this using an automated cryptographic processing system with Python scripting and FPGA implementation commands. Currently, it has an accuracy of 95% for encryption processing and uses 64-bit data packaging for cryptographic operations. We have also created a real-time visualization demonstration using eight 7-segment displays on the Nexus 4 DDR FPGA board, which shows the plain text, ciphertext, and decrypted results with user-controlled switching. The system has a response time of less than 100 milliseconds. The functionality can be validated through comprehensive testing, including mathematical verification, hardware implementation, and achieving 100% encryption accuracy for well-defined bit constraint operations. The project also supports key architectures ranging from 8-bit to 32-bit, with analysis and performance metrics for research and application purposes.