JOURNAL OF CRITICAL REVIEWS

ISSN- 2394-5125 VOL 07, ISSUE 08, 2020

Design and Analysis of a Secure Hardware Implementation for Cryptography Applications

CHETAN PANDEY,
Department of Comp. Sc. & Info. Tech., Graphic Era Hill University,
Dehradun, Uttarakhand, India 248002
DOI: 10.48047/jcr.07.08.610

Abstract:

Networks need to communicate securely to ensure that data is kept safe from unauthorized access or malicious attacks. Cryptography algorithms play a key role in achieving this goal. Symmetric key algorithms use a single key to both encrypt and decrypt data, making them fast and efficient. However, the main challenge with symmetric key algorithms is key management, as the same key needs to be securely shared among all parties involved in the communication.
On the other hand, asymmetric key algorithms use a pair of keys (a public key and a private key) to encrypt and decrypt data. The public key can be freely shared, while the private key must be kept secret. Asymmetric key algorithms are slower than symmetric key algorithms, but they offer a higher level of security. AES (Advanced Encryption Standard) is a symmetric key algorithm widely used for data encryption. It is considered to be very secure and is used by many organizations to protect sensitive data. RSA (Rivest–Shamir–Adleman) is an asymmetric key algorithm often used for key exchange and digital signatures. It is also considered to be secure, although it can be slower than other asymmetric key algorithms. By combining features of both symmetric and asymmetric key algorithms, it is possible to create a new security protocol that offers better security than using a single algorithm alone. The new protocol can leverage the speed and efficiency of symmetric key algorithms, while also benefiting from the increased security provided by asymmetric key algorithms. The main purpose of designing this new algorithm is to provide better security to data in transit against passive as well as active attacks. Passive attacks involve eavesdropping on the communication to gain access to sensitive data, while active attacks involve attempting to modify or disrupt the communication. By using a more secure protocol, organizations can protect their data from both types of attacks and ensure that it remains confidential, authentic, and available.

Introduction:

Designing a secure hardware implementation for cryptography applications requires a deep understanding of both hardware and cryptography. Here are some general steps to follow in designing and analysing such a system:
i) Identify the cryptographic algorithm(s) to be implemented: This is the first step in designing a secure hardware implementation. Select the appropriate algorithm based on the application requirements and security level desired.

ii) Determine the hardware platform: The hardware platform used to implement the algorithm should be carefully selected to ensure that it meets the performance requirements and security constraints of the application. A dedicated hardware device such as an FPGA or ASIC is usually preferred.

iii) Define the architecture: The architecture should be designed to meet the specific requirements of the cryptographic algorithm. For example, symmetric key algorithms such as AES require a different architecture than public key algorithms such as RSA. The architecture should also consider power and timing attacks.

iv) Implement the design: The design should be implemented using the selected hardware platform. Careful attention should be paid to the physical layout and interconnects to ensure that the design is secure and robust against physical attacks.

v) Test the implementation: The implemented design should be tested thoroughly to ensure that it meets the performance requirements and is secure against various attacks. Testing should include functional testing, power analysis testing, and fault injection testing.

vi) Perform a security analysis: The security of the implementation should be analyzed to ensure that it is resistant to both passive and active attacks. This includes analyzing the design against known attacks such as differential power analysis (DPA), timing attacks, and side-channel attacks.

vii) Validate the implementation: The final step is to validate the implementation against the original cryptographic specification. This ensures that the implemented design meets the desired security requirements and produces the expected output.

In summary, designing and analysing a secure hardware implementation for cryptography applications requires a thorough understanding of both hardware and cryptography. The process involves carefully selecting the cryptographic algorithm, hardware platform, architecture, implementation, testing, security analysis, and validation. A comprehensive approach is necessary to ensure that the design is secure against various attacks and produces the expected output.

Literature Review:

The paper proposes a new symmetric key cryptography algorithm for data security. The proposed algorithm is based on a combination of substitution and transposition techniques. The authors claim that their algorithm is more secure and efficient than existing symmetric key cryptography algorithms [1].

The paper provides a detailed description of each cryptographic algorithm, including its mathematical principles and the steps involved in its implementation. The authors also present the results of their evaluation, which includes a comparison of the encryption and decryption times, the key size, and the security of each algorithm [2].

The paper concludes that both symmetric key cryptography and asymmetric key cryptography have their advantages and disadvantages, and the choice of algorithm should depend on the specific requirements of the application. The authors recommend that a thorough analysis of the application's security needs and performance requirements should be conducted before choosing a cryptographic algorithm [3].

The paper concludes that the choice of block cipher algorithm should depend on the specific requirements of the application. The authors recommend that a thorough analysis of the application's security needs and performance requirements should be conducted before choosing a block cipher algorithm [4].

The paper also discusses the unique security challenges posed by cloud networks, such as multi-tenancy and virtualization, and how cryptographic algorithms can be used to address these challenges [5].

The paper discusses various implementation approaches for the Advanced Encryption Standard (AES) algorithm, which is a widely used symmetric key encryption algorithm. The authors compare and evaluate different approaches based on factors such as speed, area, power consumption, and scalability [6].

The paper concludes that the proposed algorithm is efficient and secure, and can be used for a variety of applications that require public key encryption. The authors recommend that further research be conducted to evaluate the performance and security of the proposed algorithm in different scenarios and environments [7].

The paper provides a comprehensive survey of various public key cryptosystems. The authors describe the basic principles of public key cryptography and explain how it differs from symmetric key cryptography. They also discuss the advantages and disadvantages of public key cryptography [8].

The paper proposes a new hybrid encryption algorithm that combines the RSA algorithm and the Diffie-Hellman key exchange protocol. The proposed algorithm uses RSA encryption for key exchange and Diffie-Hellman for data encryption [9].

The paper proposes a novel approach for secure communication in wireless personal area networks (WPANs) using quantum cryptography. The proposed approach is based on the BB84 protocol, which is a well-known quantum key distribution protocol [10].

The paper concludes that the LEA and LED ciphers are efficient and secure on x86 architectures, and they can be used in various applications that require lightweight cryptography, such as mobile devices and wireless sensor networks. The authors also highlight the importance of efficient lightweight cryptography in modern computing systems, where resource-constrained devices are becoming increasingly common [11].

The paper focuses on the linear cryptanalysis of reduced-round variants of the SIMECK family of block ciphers. SIMECK is a lightweight block cipher that is designed for resource-constrained devices. The author analyses the security of reduced-round variants of the cipher using linear cryptanalysis, which is a well-known cryptanalytic technique for breaking block ciphers [12].

Dedicated cryptographic hardware is designed specifically for performing cryptographic operations, and it can provide higher performance, better security, and lower power consumption compared to software implementations running on general-purpose hardware.

Dedicated cryptographic hardware can provide faster encryption and decryption speeds, especially for high-volume data transfers, such as those required for network communication or data storage. Hardware-based implementations can also reduce the likelihood of side-channel attacks, which are a type of attack that exploits unintended channels of communication in the hardware or software to extract secret information.

In addition, dedicated cryptographic hardware can provide better resistance to tampering or physical attacks, as it is designed with security in mind and can be physically protected more easily than software running on a general-purpose processor.

Difference between Hardware based cryptography and Software based cryptography:

Hardware-based cryptography refers to the use of dedicated cryptographic hardware, such as a secure microprocessor, smart card, or hardware security module (HSM), to perform cryptographic operations. The hardware is designed and built specifically for secure storage and processing of cryptographic keys and algorithms, and provides physical security against tampering and attacks.

On the other hand, software-based cryptography refers to the use of cryptographic algorithms implemented in software, usually on a general-purpose computer system or mobile device. The cryptographic keys and algorithms are stored and processed in software, and the security of the system depends on the security of the software and the device it runs on.

The main difference between the two approaches is the level of security they provide. Hardware-based cryptography offers a higher level of security because the hardware is designed specifically for secure cryptographic operations and is physically protected against attacks. In contrast, software-based cryptography is more vulnerable to attacks such as hacking and malware, as the software can be compromised or manipulated.

Another important difference is the cost and complexity of implementation. Hardware-based cryptography is generally more expensive and requires specialized hardware and expertise to implement, while software-based cryptography can be implemented on a wide range of devices and platforms with relatively low cost and complexity.

Overall, hardware-based cryptography is preferred for high-security applications such as financial transactions, while software-based cryptography is more suitable for general-purpose applications that require moderate security.

Existing Algorithms:

The Advanced Encryption Standard (AES) is an example of a symmetric key algorithm, where the same secret key is used for both encryption and decryption of data. AES is widely used in modern encryption systems due to its high level of security and efficiency.

On the other hand, RSA is an example of an asymmetric key algorithm, where two different keys are used for encryption and decryption. The public key is used for encryption, while the private key is used for decryption. RSA is commonly used for secure data transmission, digital signatures, and other cryptographic applications.

Both AES and RSA have their own strengths and weaknesses, and the choice of algorithm depends on the specific requirements of the application. It is important to carefully consider the security and performance characteristics of each algorithm when selecting the appropriate one for a given task.

Proposed Algorithm:

This new suggested encryption technique is a block cypher that acts on a block of data of identical length and then encrypts each block with a Key. This method combines the strengths (in terms of speed and security) of the RSA and AES algorithms, and it also includes certain additional security measures that increase its resistance to assaults of all kinds. Both an asymmetric key and a symmetric key are used by this technique to encrypt and decode data. The suggested approach uses 512-bit keys and a 128-bit block size with 10 rounds of encryption. The suggested approach is more secure than current algorithms because it uses two distinct encryption and decryption processes to protect the plain text.

The procedures of encryption and decryption will operate simultaneously. But because data must travel through either the encryption process or the decryption process at some point, the encryption phase will end first, followed by the decryption phase. Entering plain text is the

initial stage in this algorithm. The user may choose to utilise this basic text. The user then chooses a private key to be used for decryption along with a public key for encryption. One key generation method that automatically selects the user-selected keys was available for user assistance.

Algorithm:

- Enter Message
- Use Key Generation Algorithm
    - Encryption (Use Encryption Key -Ke)
    - Generate ciphertext 1 using phase 1 of algorithm
    - Generate ciphertext 2 using phase 2 of algorithm
- Receive ciphertext 2 at receiver
- Decryption (Use Decryption Key- Kd)
    - Generate plaintext 1 using phase 1 of algorithm
    - Generate plaintext 2 using phase 2 of algorithm

MATLAB R2019a was used to implement the experiment. In this experiment, an Intel® Core (TM) i7-7700 HQ CPU running at 2.80 GHz together with 8 GB of RAM and a 1TB HDD was employed. Three criteria were used to assess the proposed algorithm: performance, security from all sorts of attacks, and avalanche effect.

This experiment uses two separate sets of data:

1) Message 1: Alphabetic characters (CRYPTOGRAPHYPLAIN) are present.

2) Message 2: Alphanumeric character (ABCDEF0123456789F) is present.

Figure 1. represents setting of algorithms done on message 1 and message 2.

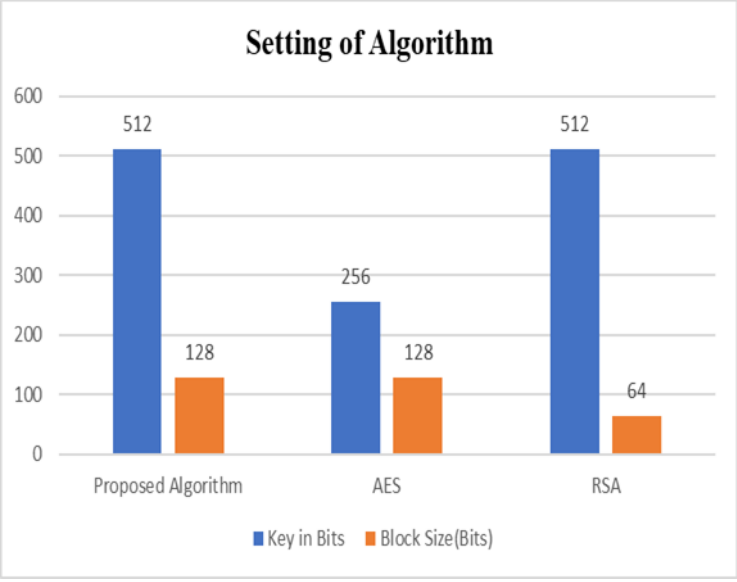

Figure 1: Setting of Algorithm

Results and Analysis:

The encryption algorithms receive the plain text as an input, and their output is an encrypted message or ciphertext. The decryption procedure was then used to this ciphertext to produce the message.

Understanding Avalanche Effect :

In cryptography, the Avalanche effect is a desirable property of cryptographic algorithms that ensures that a small change in either the input or the key results in a significant change in the output. The idea behind the avalanche effect is that if a cryptographic algorithm exhibits this property, it becomes harder for an attacker to deduce the key used in the encryption process.

In other words, if a cryptographic algorithm has a strong avalanche effect, it means that even if an attacker makes a small change to the plaintext or the key, the resulting ciphertext will look completely different from the original ciphertext. This makes it difficult for an attacker to perform statistical analysis on the ciphertext and determine the key used to encrypt the data.

The term "avalanche effect" comes from the idea that a small disturbance, like a single snowflake falling, can trigger a much larger event, like an avalanche. Similarly, in cryptography, a small change in the input or key should cause a large change in the output, creating an "avalanche" of changes throughout the ciphertext.

Overall, the avalanche effect is an important property for cryptographic algorithms, as it helps to ensure the security of the encrypted data by making it more difficult for attackers to deduce the key used in the encryption process. Figure 2. shows avalanche effect on proposed algorithm, AES and RSA.

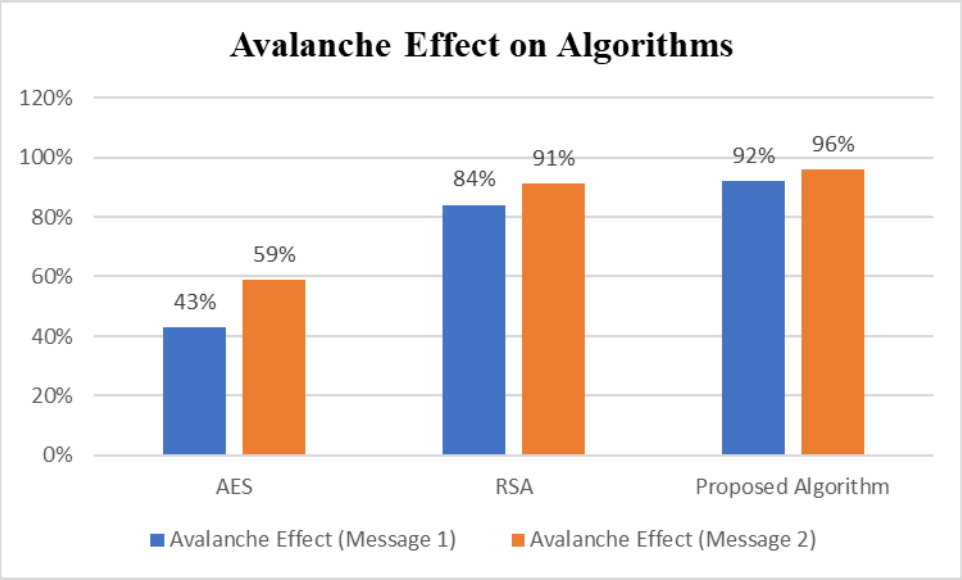

Figure 2: Avalanche Effect on Algorithms

Time Needed for Execution:

Figure 3. shows time needed for execution of message 1 and message 2 using proposed algorithm, AES, and RSA.

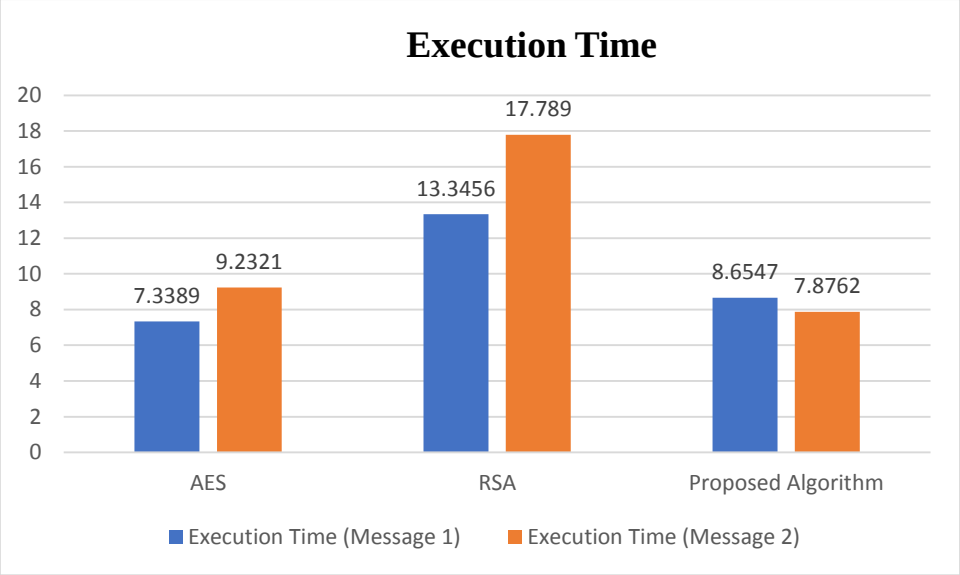

Figure 3: Time needed for Execution

Conclusion:

In real-time applications, the benefits of data encryption become obvious in the form of security and secrecy. In applications like email, e-commerce, and e-cash, where highly sensitive communication connections are utilised for the transfer of highly volatile data, encryption of data is particularly important.

Multiple factors utilised in algorithms, such as keys and plaintext block size, have been found as being important for both security and strength. Since the key length directly affects how secure an encryption technique is, the more secure the algorithm, the longer the key.

In comparison to current methods, the suggested algorithm is effective and offers more security. It performs better than AES and RSA, in comparison. Anyone who is familiar with MATLAB may do this with ease. It combines some of the properties of the AES and RSA algorithms with some other benefits, such as speed and security. Today, security is a major concern, making the suggested algorithm the ideal choice. The key aspect of the suggested approach is that the ciphertext is very hard to decipher since data is subjected to twofold encryption. This suggested technique can be used for delivering private data in business settings or for any kind of public application. Consequently, it has useful applications in the area of information security.

References:

[1] A. Anand, A. Raj, R. Kohli and V. Bibhu, “Proposed Symmetric Key Cryptography Algorithm for Data Security”, in 1st International Conference on Innovation and Challenges in Cyber Security, 2016.

[2] M. Marwaha, R. Bedi, A. Singh and T. Singh, “Comparative Analysis of Cryptographic Algorithms,” International Journal of Advanced Engineering Technology, Vol. 4, No. 3, pp. 16-18, 2013.

[3] S. Chandra, S. Paira, S. S. Alam and G. Sanyal, "A comparative survey of symmetric and asymmetric key cryptography," in International Conference on Electronics, Communication and Computational Engineering, Nov 2014.

[4] I. Alam, and M. E. R. Kahn, "Performance and Efficiency Analysis of Different Block Cipher Algorithms of Symmetric Key Cryptography," International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3, No. 10, pp. 713-720, 2013.

[5] V. K. Singh and M. Dutta, "Analysing Cryptographic Algorithms for Secure Cloud Network," International Journal of Advanced Studies in Computer Science and Engineering, Vol. 3, No. 4, pp. 1-9, 2014.

[6] X. Zhang and K. K. Parhi, "Implementation Approaches for the Advanced Encryption Standard Algorithm," IEEE Circuits and Systems Magazine, Vol. 2, No. 4, pp. 24 – 46, 2002.

[7] D. Pugila, H. Chitrala, S. Lunawat and P. M. D. R. Vincent," An Efficient Encryption Algorithm Based on Public Key Cryptography," International Journal of Engineering and Technology, Vol. 5, No. 3, pp. 3064-3067, 2013

[8] A. Ganpati and N. Tyagi," A Survey of Different Public-Key Cryptosystems," International Journal of Computer Science Trends and Technology, Vol. 3, No. 6, pp. 66-70, 2015.

[9] S. Gupta and J. Sharma, "A Hybrid Encryption Algorithm based on RSA and Diffie-Hellam," in IEEE International Conference on Computational Intelligence and Computing Research, 2012.

[10] S. Nagpal, "Quantum Cryptography Integrated Effective Communication Approach for WPAN," International Journal of Enhanced Research in Management & Computer Applications, Vol. 5, No. 9, pp. 1-5, Sept 2016.

[11] R. Benadjila, J. Guo, V. Lomne, and T. Peyrin, "Implementing ´ lightweight block ciphers on x86 architectures," in Proceedings of the Selected Areas in Cryptography (SAC), Lecture Notes in Computer Science, vol. 8282, Springer, Berlin, Heidelberg, August 2013.

[12] N. Bagheri, "Linear cryptanalysis of reduced-round SIMECK variants," in Proceedings of the Progress in Cryptology- –INDOCRYPT 2015, pp. 140–152, Springer, Bangalore, India, December 2015.