Encryption Algorithm

Nur Muhammad [1] Gourab Debnath Himel [2] Md: Monowar Hossain Parves[3] Sumit Saha Swapno [4] Fatema Begum Maliha [5]

Department of Computer Science and Engineering
Independent University, Bangladesh
Dhaka, Bangladesh.
{[1]2220410,[2]2220577,[3]2221376,[4]2220207,[5]2220556}@iub.edu.bd

Abstract

The Advanced Encryption Standard(AES) is a symmetric block cipher chosen by the US government to cover classified information. Joan Daemen and Vincent Rijmen construct the AES algorithm. It was published on November 26, 2001. The National Institute of morals and Technology(NIST) started the development of AES. It announced the need for volition to the Data Encryption Standard(DES), which was starting to come vulnerable to brute- forceattracts.AES data encryption is a more mathematically effective and elegant cryptographic algorithm, but its main strength rests in the option for various pivotal lengths. AES allows you to choose a 128- bit, 192- bit or 256- bit pivotal, making it exponentially stronger than the 56- bit crucial of DES.The data we store on Google Drive is an illustration of the operation of the AES algorithm. The pall on which the user data is stored and visible on Google uses the AES encryption system. It deploys a 256- bit encryption system, which is considered a more complex and largely secured system.

Introduction

Encryption algorithms are a fine formula that, with the help of a key, changes plaintext into ciphertext. They also make it possible to reverse the process and revert ciphertext into plaintext. Stylish Encryption Algorithms textbf AES, Blowfish, Twofish, Rivest- Shamir- Adleman(RSA). AES is a symmetric encryption algorithm, which means that the same key is used for both encryption and decryption. This key participates between the sender and the receiver of the translated data, and it must be kept secret to ensure the security of the data. The RSA algorithm is a public-crucial hand algorithm developed by Ron Rivest, Adi Shamir, and Leonard Adleman. Their paper was first published in 1977, and the algorithm uses logarithmic functions to keep the working complex enough to repel brute force and streamlined enough to be the fastest-deployment. Blowfish is a symmetric master cipher system that depends on the Feistel network. Bruce Schneider introduced the algorithm. It's a 64- bit block size cipher, and the complete interpretation needs 16 rounds to complete the block cipher and uses a high number of subkeys, and variable-length keys from 32- bits to 448- bits. Twofish is a symmetric-crucial block cipher with a block size of 128 bits and variable-length key of size 128, 192, or 256 bits. This encryption algorithm is optimized for 32-bit central processing units and is ideal for both tackle and software surroundings. We elect textbf AES encryption algorithms.

Rationale for the Algorithm's Selection

We select **AES** because The Advanced Encryption Standard is a symmetric encryption algorithm that's the most constantly used system of data encryption encyclopedically. Frequently appertained to as the gold standard for data encryption, AES is used by numerous government bodies worldwide, including in the U.S. AES focuses on four steps;

1. Byte substitution: This step replaces each byte in the block with a different byte, based on a lookup table.
2. Shift rows: This step shifts the rows of the block by a certain number of positions.
3. Mix columns: This step mixes the columns of the block using a mathematical formula.
4. Add round key: This step adds the block to a round key, which is a randomly generated string of bits.

Methodology

The encryption process in AES starts with the original data, called plaintext, which is added to a randomly generated string of bits known as a round key. This step is called AddRoundKey. The next step involves performing a series of transformations on the data to make it difficult to decipher without the correct key. The transformations used in AES include SubBytes, which replaces each byte of data with a different byte based on a lookup table, ShiftRows, which shifts the rows of data by a certain number of positions, and MixColumns, which mixes the columns of data using a specific formula. These transformations are repeated several times, depending on the number of rounds, which is either 10 or 12 depending on the key size. After the final round, the data is added to the final round key, resulting in the ciphertext, which is the translated data that can be transferred over a network or stored on a device. To decrypt, the ciphertext is added to the first round key, and the transformations are performed in reverse order, resulting in the original plaintext.

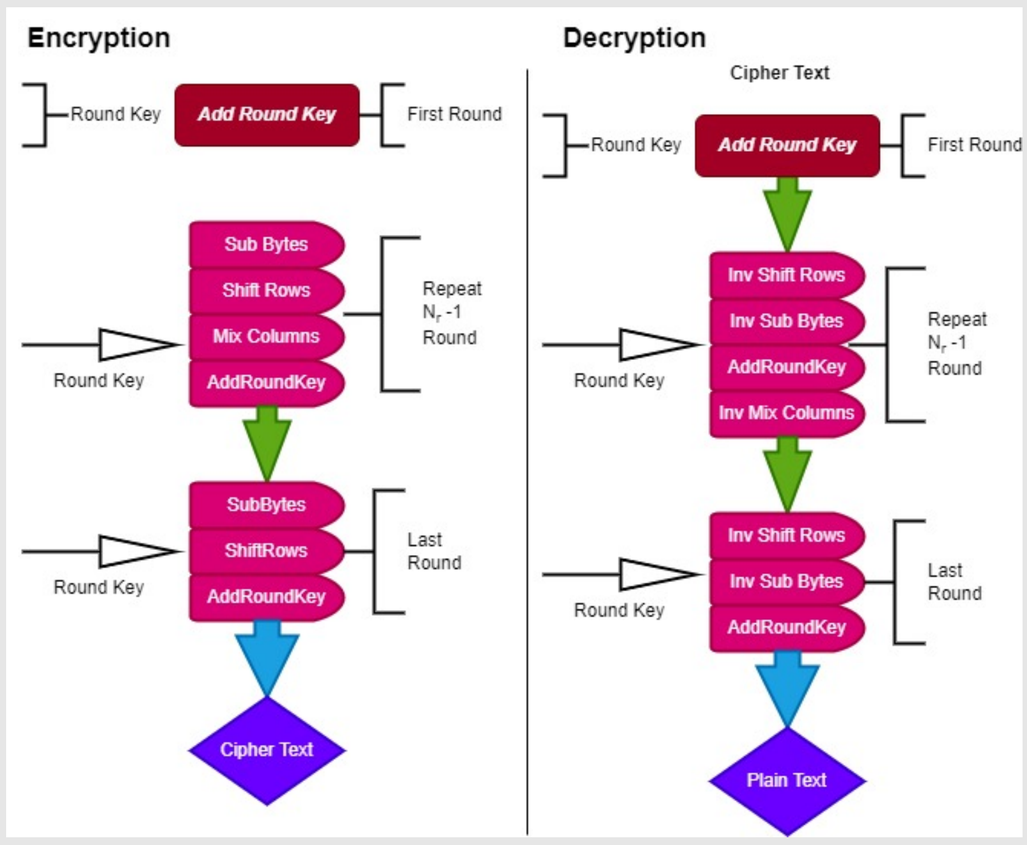

Figure 1. **Flowchart**

Substitute Use of the Algorithm

An algorithm is a fundamental power in computer science and problem-solving. Many algorithms can be adapted or reused for various applications beyond their original context. Here's how and why this can happen:

1.General Problem-Solving Techniques: Algorithms often present general problem-solving techniques that are not confined to specific domains. For example, algorithms like QuickSort or MergeSort offer selection methods useful in different situations where arranging elements is needed, such as sorting data, task scheduling, or process optimization.
2.Abstraction: Algorithms can be abstracted and described at a higher level, focusing on logical steps rather than specific data or contexts. This abstraction allows applying the algorithm in diverse situations with minimal changes. For instance, graph algorithms like Dijkstra's can be employed in various networks such as transportation systems, social networks, and computer networks.
3.Data Transformation: Algorithms that manipulate data can often be tailored to different types of data. Encryption algorithms, for example, can secure communication beyond cybersecurity, like protecting financial transactions or personal information.
4.Optimization and Search Algorithms: Algorithms designed to optimize a function or search for patterns can find applications in different areas. Genetic algorithms, inspired by natural evolution, can optimize complex systems like supply chains, asset allocation, and creative design processes.
5.Machine Learning Algorithms: Many machine learning algorithms, such as neural networks, decision trees, and support vector machines, learn from data patterns. These algorithms can be fine-tuned for various domains by training with relevant datasets.
6.Inspiration and Analogies: Algorithms can inspire solutions in unrelated domains. Nature-inspired algorithms like genetic algorithms, particle swarm optimization, or ant colony optimization have optimized complex systems in engineering, economics, and logistics.
7.Evolution and Refinement: As technology advances, algorithms can be refined and adapted for emerging technologies. Algorithms developed for classical computers can be reimagined for quantum computing as the field progresses.

Alternative Solution

Yes, in many cases, algorithms can be replaced with other similar or alternative algorithms to achieve better overall performance. The decision to replace an algorithm depends on various factors, including the specific problem, the characteristics of the data, the available resources, and the goals of optimization. Here are some scenarios where replacing an algorithm might lead to better performance:

1.Efficiency and Speed: If an alternative algorithm has a better time complexity (faster execution) or space complexity (uses less memory) than the current algorithm, it may lead to improved performance. For example, replacing a brute-force search algorithm with a more efficient search algorithm like binary search can significantly speed up the process.
2.Accuracy and Precision: In some cases, a more advanced or specialized algorithm might offer improved accuracy or precision. For instance, replacing a simple linear regression algorithm with a more sophisticated machine learning algorithm like a random forest or gradient boosting can lead to better predictive accuracy.
3.Scalability: If the problem's scale increases, an algorithm that scales better with larger datasets might be more suitable. Algorithms with better scalability can handle larger input sizes while maintaining acceptable performance. This is crucial in modern applications dealing with big data.
4.Resource Utilization: If an alternative algorithm requires fewer computational resources, such as CPU time or memory, it could lead to better utilization of available resources, allowing for more efficient multitasking or cost savings in cloud computing environments.
5.Parallelism and Concurrency: Some algorithms are better suited for parallel processing or concurrent execution. If your problem can be divided into parallel tasks, switching to an algorithm optimized for parallelism can lead to significant speedup on multi-core processors or distributed computing environments.
6.Domain-Specific Considerations: Some algorithms are specifically designed for certain domains or types of data. If your problem fits the characteristics of a specialized algorithm, it could provide better results compared to a more general-purpose algorithm.
However, it's important to note that replacing an algorithm isn't always a straightforward process. The benefits of switching to a different algorithm need to be balanced against potential challenges such as implementation complexity, code maintenance, and potential unintended consequences. Thorough testing and benchmarking are crucial before making any major algorithmic changes to ensure that the expected performance gains are realized in practice. I'd be delighted to assist in suggesting changes to an existing system, but I'd need more precise details on the system in issue. Please describe the present system, its purpose, the issues you're experiencing, and any particular aims you have for change. The more information you share, the more I can adapt my recommendations to your specific requirements.

Conclusion

We conclude this section, the AES algorithm computes much faster than RSA in execution and implementation. RSA algorithm is reliable for key exchange management but it's not very efficient in terms of performance and cost factor. RSA's strengths and weaknesses remain in the factoring large integers.