Comparing Algorithms and Their Performance Tradeoffs in Encrypting and Decrypting Images Across a Computer Network

Nathan Neeley
Department of Computer Science
Kennesaw State University Marietta, GA
nneeley@students.kennesaw.edu

ABSTRACT

In recent years, there has been extensive research on encryption algorithms and how they are used and how to improve them. Past research has considered comparing each algorithm and determining the tradeoffs of using each one. I aim to take more of a comprehensive survey of encryption algorithms by considering several aspects for comparison in encrypting/decrypting images. The focal point of this research is to determine which algorithm is best suited for a specific circumstance. This investigation should yield advantageous results for anyone that is working with encrypting data that would like to make the best algorithmic choice possible.

KEYWORDS

Cryptography, Encryption, Decryption, Computer Networking, Data Computing

1 INTRODUCTION

In recent years, there has been extensive research on encryption algorithms and how they are used and how to improve them. There are many encryption algorithms, but I will only consider the ones that are most widely used: RSA, DES, 3DES, and AES. Each algorithm is geared towards either performance or greater security. Past research has considered comparing each algorithm and determining the tradeoffs of using each one. Additionally, most research deals with encrypting text or bytes, but my analysis will consider only images. I aim to take more of a comprehensive survey of encryption algorithms by considering several aspects for comparison in encrypting/decrypting images. First, I will discuss how each algorithm works to encrypt/decrypt an image. Second, I will develop an implementation in Python for each algorithm, focusing on properties of each such as time, space, and security tradeoffs. This implementation section will be the cornerstone of this research. Third, the focal point of the implementation is to analyze which algorithm is ideal for a specific circumstance. In other words, one algorithm might be better used with more images when performance is a priority, whereas another algorithm might be better when only a small number of images need to be encrypted at a time, but security is the most important factor. This investigation should yield advantageous results for anyone that is working with encrypting data that would like to make the best choice possible when picking an algorithm.

2 LITERATURE REVIEW

There has been an extensive corpus of research done on encryption algorithms and comparing them. Among them, the large majority have dealt with the most common published algorithms, which include RSA, DES, 3DES, and AES, but some have expanded on these algorithms, devising new algorithms and modifying parts of the preexisting common algorithms to improve performance or security. The following

section will consider some of the research papers on a multitude of encryption algorithms.

[1] compares DES, 3DES, and AES to explain their purposes and compare their performances. DES has been determined to be vulnerable to brute attack and be inferior to AES, but this paper also considers 3DES, which was a product of DES being too prone to attack. The paper concluded that AES has advantage over 3DES for speed and efficiency in some hardware implementations, but 3DES may be faster where support for 3DES has matured.

[2] compares encryption algorithms, which include DES, 3DES, AES, and Blowfish, to determine the best one to use. It states that DES is still the most widely used block cipher in the world despite AES considered to offer greater security and performance. After analyzing results, even though Blowfish is lesser used than the other algorithms, Blowfish was determined to be the fastest algorithm even though security was not considered.

[3] gives us a starting point for analyzing some of the tradeoffs of using different encryption algorithms. It walks through how each algorithm works and the time and space performance when encrypting documents of varying size. It compares the results and determines that AES performed better than DES and RSA algorithms.

3 PROPOSED SOLUTION

For my proposed solution to expand the prior research corpus, I applied encryption algorithms (RSA, DES, 3DES, and AES) to datasets of images (small and large) to determine their performance. Previous research dealt with analyzing a single document or text of varying size whereas mine encrypts full datasets to see the larger performance differences. I also chose images because they have a dimensionality aspect as opposed to text which is one dimension.

3.1 Problems. One problem is in developing an implementation for each algorithm that is optimized, so I can make a proper comparison when analyzing the tradeoffs. Furthermore, to make this research relevant and beneficial, I need to make sure enough properties are considered for each algorithm, so I can be sure the accuracy of determining which algorithm is the best for a specific circumstance. If components are not included that are important when examining time, space, or security, then the ideal circumstance for using the algorithm is potentially inaccurate, and thus this research is not comprehensive and is ultimately obsolete.

4 IMPLEMENTATION

Here I will walk through how I developed my implementation and briefly explain how each encryption algorithm works that I am using to give prerequisite information for the evaluation and results section to be clear.

I implemented my project in Python, and I used the built-in libraries (rsa, des, and aes) since there are nuances and a lot of complexity in implementing the algorithms from scratch, and by using the python's libraries, it is more likely that the algorithms are optimized and have been thoroughly tested. Since the performance and security is similar during the encryption and decryption processes, I focused on the encryption and disregarded the decryption.

Below is the pseudocode for how I applied each of the encryption algorithms in my code:

Pseudocode:

```
Set key(s) for encryption/decryption
While there are images left to encrypt
    While there are blocks left in image
        Encrypt(block)
        Add block to image
    Add encrypted image to list
```

4.1 Encryption Algorithms

4.1.1 RSA. The RSA algorithm is an asymmetric encryption algorithm. It is the most popular algorithm of this type still in use. It is referred to as an asymmetric algorithm because it uses two keys, a public key and a

private one. The benefit of this type of algorithm is that the public key is used for encryption and is then sent across a secure network while the private key is used in decryption. Even if the public key is intercepted, the private key is kept isolated from the computer networking stream, making it impossible to decrypt the data. The strength of RSA relies on the fact that it is difficult to factorize large prime numbers. The public and private keys are derived from two large prime numbers. Therefore, if the large number is factorized, the private key is compromised. Furthermore, by doubling or tripling the key size, the effectiveness of the algorithm is improved exponentially, typically generating keys of 1024 or 2048 bits long.

Below are the steps for generating the public and private keys and how to encrypt/decrypt the data:

Generating Public Key:

1. Select two prime numbers (p and q)
2. Generate n. n = p*q
3. Select small exponent (e)
4. Generate phi(n). phi(n) = (p-1)(q-1)
5. Ensure e and phi(n) are not coprimes
6. Ensure 1 < e < phi(n)
7. Public key encryption is $C = P^e \bmod n$

Generating Private Key:

1. Calculate phi(n). phi(n) = (p-1)(q-1)
2. Calculate d. d = (k*phi(n) + 1) / e for some integer k
3. Private key decryption is $P = C^d \bmod n$

4.1.2 Data Encryption Standard (DES). DES is a block cipher and encrypts data of 64-bits at a time, using a key size of 56-bits. The initial key size if 64-bits, but every 8th bit is discarded to produce a 56-bits key. This algorithm is based on fundamental aspects of cryptography, referred to as substitution and transposition. Before the encryption process begins, an initial permutation is performed. First, the 58th bit is replaced with the 1st bit. Second, the 50th bit is replaced with the 2nd bit, and so on, moving down by 8 bits each time until all permutations are completed. After this, the 64-bits block are divided into two 32-bits halves in which each of the 16 rounds of the algorithm is performed on it before finally combining back together to perform one final step to complete the encryption process.

Below are the steps for each of the 16 rounds:

1. Key transformation
2. Expansion permutation
3. S-box permutation
4. P-box permutation
5. XOR and swap

Before the encryption process is complete, the 32-bits halves are combined into a 64-bits block. Then the final permutation is performed, which is the inverse of the initial permutation before the 16 rounds.

4.1.3 Triple Data Encryption Standard (3DES). Since DES was determined to be susceptible to brute force attacks and other security attacks because of the key size only being 56-bits, 3DES was created to counteract DES's security vulnerabilities, increasing the key size to 168 bits (3*56). 3DES is more involved than just increasing the key size and performing the steps three times the number of times.

Below are the steps for 3DES:

1. Encrypt the plaintext with single DES using key 1 (first 56-bits)
2. Decrypt the output of step 1 with single DES using key 2 (second 56-bits)
3. Encrypt the output of step 2 with single DES using key 3 (second 56-bits)
4. The encryption data is the output of step 3

If key 1, key 2, and key 3 are set to the same value, a single DES implementation is performed three times, providing backward compatibility with DES. Clearly 3DES improves DES's security vulnerabilities, but adds significant performance time to the process, forcing us to measure the tradeoffs between security

and performance when deciding on an encryption algorithm.

4.1.4 Advanced Encryption Standard (AES). AES was created based on a contest in the late 1990s to develop the best encryption algorithm, which was a result of DES being prone to a brute attack because of the key size. In 2001 AES was adopted and is now the standard for secure encryption due to its key size, security, and performance. However, it does offer more complexity than DES, being more difficult when implementing. AES is a block cipher, encrypting with key sizes of 128, 192, or 256 bits. The data is encrypted in blocks of 128-bits. The algorithm encrypts data in bytes rather than bits, which is one reason why performance is better than other algorithms. Depending on the size of the key, the number of rounds (10, 12, or 14 rounds) of the encryption process changes. Prior to the encryption steps, a key schedule algorithm is used to calculate all the round keys from the initial key.

Below are the steps of each round of AES:

1. Bytes Substitution
2. Rows Shifting
3. Columns Mixing (not included in last round)
4. Output of step 3 (or step 2 for last round) is XOR-ed with round key

To date, we have still failed to break AES encryption, leaving its only security vulnerability to its implementation.

5 EVALUATION AND RESULTS

I performed each encryption algorithm on the small and large datasets to compare their results. By applying the algorithm to full datasets, we can see the overarching differences between their performances. Below is the table showing their performance times:

Encryption Algorithm	**Small Dataset Performance (300)**	**Large Dataset Performance (20580)**
RSA	6.103 Secs	312.496 Secs
DES	734.954 Secs	135078.68 Secs
3DES	196062.66 Secs	Und.
AES	0.190 Secs	178.947 Secs

As you can see from the table, AES outperformed its counterparts by a huge margin in terms of performance. In comparison, RSA performed at a reasonable speed on both the small and large datasets. As we already discussed, RSA is a relatively secure encryption algorithm to use for a few reasons. It is built around factoring large prime numbers, which is difficult to crack for a computer. Another benefit is the use of a public and a private key. Even if the public key becomes compromised, the private key still remains secure. The advantage over AES is in the simplicity of it. AES contains many more stages and parts in the implementation construction, whereas RSA is relatively simple to implement. The performance might be slightly slower, but simplicity might be what one is seeking out that employs RSA. From the table, DES and 3DES are slow enough that one would be hard pressed to choose one of these algorithms. As we already discussed, with the speed of current computers, DES encryption has been broken, so it is unwise to choose this algorithm if maximum security is the goal. Interestingly though, this algorithm is still the most popular in production out there, possibly due to efficacy of it when it came out. 3DES does offer increased security over DES even though performance takes a hit, especially since 3DES employs three keys, exponentially increasing the time that a brute-force attack would take to break the encryption. The performance of 3DES on the large dataset was too long for it to even complete within a reasonable period, hence why I labelled it undefined in the table. Even though 3DES is typically much slower than its counterparts and the security features do not outweigh the advantages of employing AES, we learned in the

literature review that some hardware implementations might be beneficial to use in some cases [1]. This theory makes sense because DES was originally created for hardware implementations and not software ones.

6 CONCLUSIONS

After analyzing the performance results of the four encryption algorithms, it is clear that AES performs the best on the small and large datasets. RSA could offer enough security and needs depending on the circumstance due to the use of a public and private key and the simplicity of implementing this algorithm over other ones. Even though the literature review determined that 3DES could be faster and offer greater security because of the added complexity in some hardware implementations, AES has proved in this circumstance to be advantageous over the other algorithms. For future work, it would be beneficial to measure performance of other lesser-known encryption algorithms to determine if they offer a performance advantage since the Blowfish algorithm already proved to do so.

REFERENCES

[1] Alanazi, H., Zaidan, B. B., Zaidan, A. A., Jalab, H. A., Shabbir, M., & Al-Nabhani, Y. (2010). New comparative study between DES, 3DES and AES within nine factors. *arXiv preprint arXiv:1003.4085*.

[2] Nadeem, A., & Javed, M. Y. (2005, August). A performance comparison of data encryption algorithms. *In 2005 international Conference on information and communication technologies* (pp. 84-89). IEEE.

[3] Padmavathi, B., & Kumari, S. R. (2013). A survey on performance analysis of DES, AES and RSA algorithm along with LSB substitution. *IJSR, India*, 2, 2319-7064.