DSN4092 - Capstone Project Phase - 2 Review - 3

CRYPTOGRAPHY WITH INFORMATION THEORY

By: Group-100

Supervised by: Dr.Mohammad Sultan Alam

Assistant Professor

School of Computer Science and Engineering

VIT BHOPAL UNIVERSITY

Group Members:

Abhishek Kumar	21BCE10084
Karan Kumar Chauhan	21BCE10243
Deep Radadiya	21BCE11343
Vikash Kumar Sinha	21BCE11500
Amul Gupta	21BCE11606

Contents:

Introduction

- In today’s digital world, securing communication is vital. With rising cyber threats, strong encryption is essential to protect sensitive data. Cryptographic algorithms rely on the strength and unpredictability of their keys, where randomness plays a key role. Predictable keys are vulnerable to attacks, making Shannon entropy a crucial measure of key strength—higher entropy means more secure, unpredictable keys.
- This study compares the key randomness and entropy of popular symmetric algorithms (AES, 3DES, DES, Blowfish) and asymmetric ones (RSA, DSA, ECC). It evaluates each based on entropy per byte, randomness, and computational efficiency.
- The results reveal trade-offs between entropy, key size, and performance, emphasizing the importance of choosing algorithms based on entropy for better security. These insights are especially relevant for securing cloud systems, IoT, mobile platforms, and enterprise networks. By exploring the link between randomness and encryption strength, this research helps improve defenses against evolving cyber threats.

Motivation

- As cyber threats grow in sophistication, ensuring the security of cryptographic systems is more critical than ever. The strength of encryption largely depends on the randomness of key generation, which affects resistance to brute-force and cryptanalytic attacks.
- Entropy, a measure of unpredictability, is key to evaluating cryptographic strength. Higher entropy means stronger, less predictable keys, while lower entropy can expose vulnerabilities. With attackers gaining more computational power, understanding how encryption algorithms handle entropy is crucial for secure system design.
- This research analyzes the entropy and key randomness of widely used symmetric and asymmetric algorithms, offering insights into their security strength, efficiency, and implementation trade-offs for modern applications.

Objective

1. Evaluate Key Randomness: Assess the level of unpredictability in cryptographic keys generated by AES, 3DES, DES, Blowfish, RSA, DSA, and ECC.

2. Measure Entropy Levels: Utilize Shannon entropy as a statistical metric to quantify the randomness and security strength of encryption keys.

3. Compare Symmetric and Asymmetric Algorithms: Examine the differences in entropy per byte, key size, and security efficiency between symmetric and asymmetric encryption techniques.

4. Assess Security Implications: Identify how entropy levels impact cryptographic strength and resistance to brute-force and cryptanalytic attacks.

5. Provide Practical Insights: Offer recommendations on selecting encryption algorithms based on their entropy properties, computational efficiency, and security requirements for diverse applications, including resource-constrained environments.

- Existing work/Literature Review

- **Symmetric Encryption and Entropy Studies**

Several studies have explored the entropy levels of symmetric encryption algorithms such as AES, 3DES, DES, and Blowfish. Research indicates that 3DES exhibits higher entropy due to its multi-layered encryption process, while AES and Blowfish also demonstrate strong randomness characteristics. However, DES has been identified as inadequate for modern security applications due to its lower entropy levels and susceptibility to brute-force attacks. Prior work highlights the necessity of longer key lengths and robust randomness sources to enhance symmetric encryption security.

- **Asymmetric Cryptography and Entropy Analysis**

The entropy characteristics of asymmetric encryption schemes, including RSA, DSA, and ECC, have been widely studied. Due to their reliance on mathematical structures, these algorithms generally exhibit lower entropy per byte. However, researchers have shown that larger key sizes compensate for this limitation, ensuring strong security. ECC has been recognized for its efficiency in providing robust encryption with smaller key sizes, making it a suitable choice for resource-constrained environments like IoT devices and mobile applications.

- **Shannon Entropy in Cryptographic Analysis**

Shannon entropy serves as a key metric for evaluating randomness in cryptographic systems. Existing literature emphasizes its significance in assessing the unpredictability of keys, with studies demonstrating that higher entropy values correspond to stronger security. Researchers have employed programmatic approaches to analyze entropy distribution across different cryptographic algorithms, reinforcing the importance of key randomness in preventing cryptanalytic attacks.

- **Symmetric Encryption and Entropy Studies**

Several studies have explored the entropy levels of symmetric encryption algorithms such as AES, 3DES, DES, and Blowfish. Research indicates that 3DES exhibits higher entropy due to its multi-layered encryption process, while AES and Blowfish also demonstrate strong randomness characteristics. However, DES has been identified as inadequate for modern security applications due to its lower entropy levels and susceptibility to brute-force attacks. Prior work highlights the necessity of longer key lengths and robust randomness sources to enhance symmetric encryption security.

- **Asymmetric Cryptography and Entropy Analysis**

The entropy characteristics of asymmetric encryption schemes, including RSA, DSA, and ECC, have been widely studied. Due to their reliance on mathematical structures, these algorithms generally exhibit lower entropy per byte. However, researchers have shown that larger key sizes compensate for this limitation, ensuring strong security. ECC has been recognized for its efficiency in providing robust encryption with smaller key sizes, making it a suitable choice for resource-constrained environments like IoT devices and mobile applications.

- **COMPARATIVE STUDIES ON CRYPTOGRAPHIC SECURITY**

PRIOR RESEARCH HAS EXTENSIVELY COMPARED SYMMETRIC AND ASYMMETRIC ENCRYPTION TECHNIQUES, EVALUATING THEIR TRADE-OFFS IN TERMS OF SECURITY, ENTROPY, AND COMPUTATIONAL EFFICIENCY. FINDINGS SUGGEST THAT WHILE SYMMETRIC ALGORITHMS PROVIDE FASTER ENCRYPTION AND DECRYPTION, ASYMMETRIC ENCRYPTION OFFERS ENHANCED SECURITY THROUGH COMPLEX KEY STRUCTURES AND PUBLIC-PRIVATE KEY MECHANISMS. RECENT ADVANCEMENTS IN CRYPTOGRAPHIC RESEARCH CONTINUE TO FOCUS ON OPTIMIZING ENTROPY WHILE BALANCING COMPUTATIONAL PERFORMANCE.

- **SHANNON ENTROPY IN CRYPTOGRAPHIC ANALYSIS**

SHANNON ENTROPY SERVES AS A KEY METRIC FOR EVALUATING RANDOMNESS IN CRYPTOGRAPHIC SYSTEMS. EXISTING LITERATURE EMPHASIZES ITS SIGNIFICANCE IN ASSESSING THE UNPREDICTABILITY OF KEYS, WITH STUDIES DEMONSTRATING THAT HIGHER ENTROPY VALUES CORRESPOND TO STRONGER SECURITY. RESEARCHERS HAVE EMPLOYED PROGRAMMATIC APPROACHES TO ANALYZE ENTROPY DISTRIBUTION ACROSS DIFFERENT CRYPTOGRAPHIC ALGORITHMS, REINFORCING THE IMPORTANCE OF KEY RANDOMNESS IN PREVENTING CRYPTANALYTIC ATTACKS.

- **FRONT-END, BACK-END, AND SYSTEM REQUIREMENTS FRONT END**

KEY FEATURES OF THE FRONTEND:

1. COMMAND-BASED INTERACTION:
   - USERS INPUT PLAINTEXT MESSAGES AND PARAMETERS DIRECTLY IN JUPYTER NOTEBOOK CELLS.
   - FUNCTIONS ARE EXECUTED STEP BY STEP, ALLOWING USERS TO OBSERVE THE ENCRYPTION PROCESS IN REAL TIME.
2. DATA VISUALIZATION TOOLS:
   - ENTROPY GRAPHS: THE SYSTEM CALCULATES AND PLOTS SHANNON ENTROPY VALUES TO ANALYZEINFORMATION SECURITY.
   - PROBABILITY DISTRIBUTIONS: VISUALIZATION OF KEY RANDOMNESS USING HISTOGRAMS.
   - ENCRYPTION PROCESS DEMONSTRATION: STEP-BY-STEP REPRESENTATION OF ENCODING, TRANSMISSION, AND DECODING.
3. INPUT & OUTPUT HANDLING:
   - TEXT-BASED INPUT ALLOWS USERS TO ENTER PLAINTEXT MESSAGES FOR ENCRYPTION.
   - OUTPUT DISPLAYS ENCRYPTED CIPHERTEXT, DECRYPTION RESULTS, AND SECURITY METRICS.
   - INTERACTIVE DEBUGGING WITH IMMEDIATE FEEDBACK FROM EXECUTED PYTHON CELLS

BACKEND:

THE BACKEND IS RESPONSIBLE FOR PERFORMING ALL CRYPTOGRAPHIC OPERATIONS, MANAGING KEY EXCHANGES, AND ENSURING SECURE COMMUNICATION. IT LEVERAGES PYTHON'S COMPUTATIONAL POWER TO IMPLEMENT SECURITY ALGORITHMS BASED ON INFORMATION THEORY.

KEY FUNCTIONS OF THE BACKEND:

1. MATHEMATICAL COMPUTATION AND SECURITY ANALYSIS:
   - SHANNON ENTROPY CALCULATIONS DETERMINE THE RANDOMNESS AND SECURITY LEVEL OF MESSAGES. PROBABILITY AND STATISTICAL FUNCTIONS ARE USED FOR KEY GENERATION AND VERIFICATION.
2. ENCRYPTION & DECRYPTION ALGORITHMS:
   - IMPLEMENTATION OF CUSTOM ENCRYPTION TECHNIQUES BASED ON INFORMATION THEORY.
   - USE OF PYTHON'S PYCRYPTODOME LIBRARY FOR ADDITIONAL SECURITY MECHANISMS.
3. SECURE KEY EXCHANGE:
   - GENERATION OF SECURE KEYS USING RANDOMNESS-BASED CRYPTOGRAPHIC TECHNIQUES.
   - KEY DISTRIBUTION SIMULATED WITHIN JUPYTER NOTEBOOK WITHOUT EXTERNAL DEPENDENCIES.
4. DATA TRANSMISSION SIMULATION:
   - THE SYSTEM SIMULATES THE SECURE TRANSMISSION OF ENCRYPTED MESSAGES.
   - DEMONSTRATES POTENTIAL ATTACK SCENARIOS AND DEFENSES

SYSTEM REQUIREMENTS

To run this system smoothly, certain software and hardware requirements must be met.
Software Requirements:

· Operating System: Windows, Linux, or macOS
· Programming Language: Python 3.x (Recommended: Python 3.8 or later)

· Required Python Libraries:

- numpy – Numerical computations
- scipy – Statistical functions
- sympy – Symbolic mathematics
- matplotlib – Data visualization
- seaborn – Advanced plotting
- pycryptodome – Cryptographic operations
- pandas – Structured data handling

Hardware Requirements

· Minimum Configuration:

- Processor: Intel Core i3 / AMD equivalent
- RAM: 4GB
- Storage: 5GB free disk space
- Internet: Required for installing dependencies

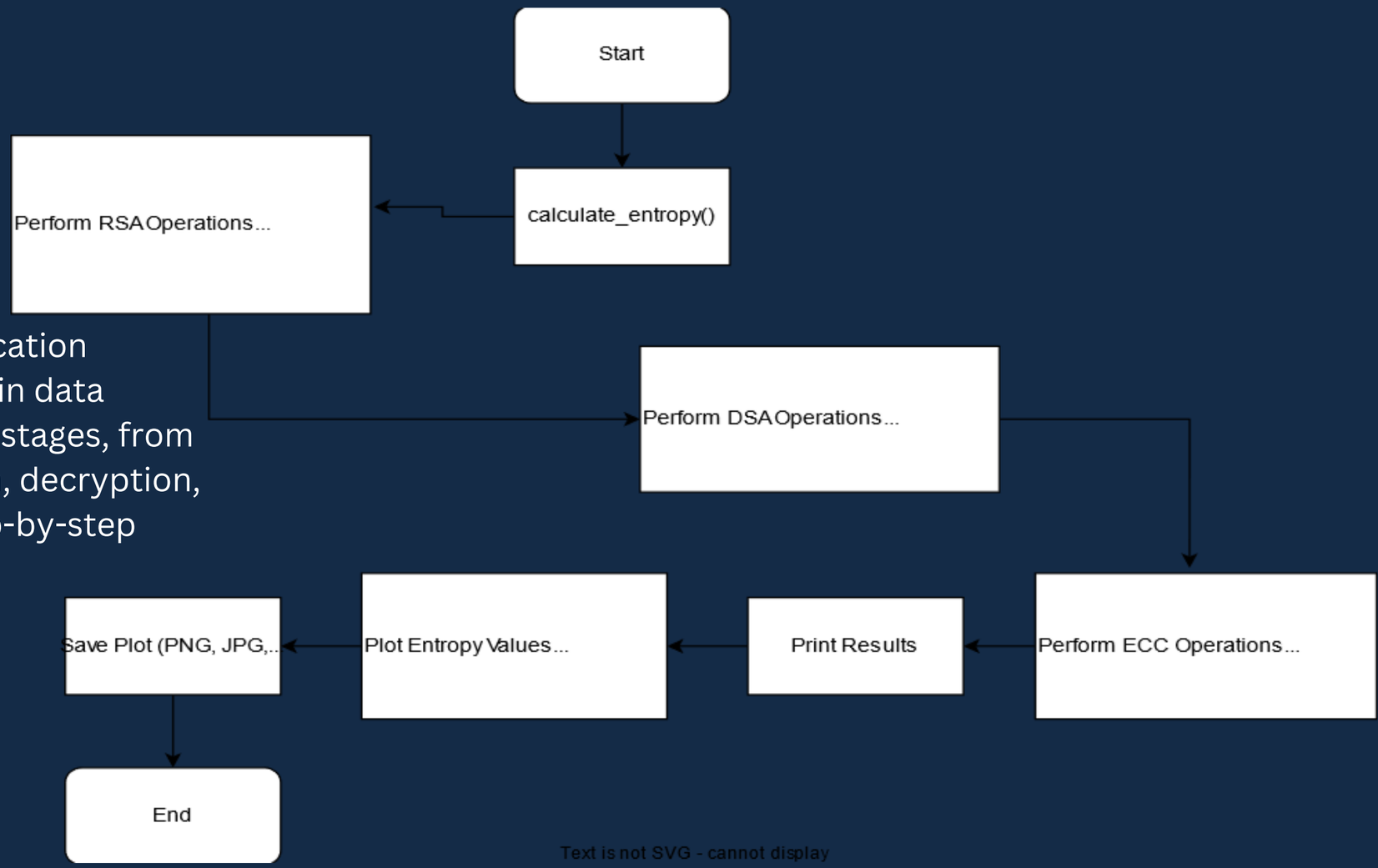
· Recommended Configuration for Better Performance:

- Processor: Intel Core i5/i7 or AMD Ryzen
- RAM: 8GB+
- Storage: SSD for faster execution

METHODOLOGY

System Design / Architecture

The system follows Shannon’s secure communication model, ensuring confidentiality and robustness in data transmission. The workflow consists of multiple stages, from message generation to encryption, transmission, decryption, and security evaluation. Below is a detailed step-by-step breakdown of the system’s working.



1. Message Generation (Source)

· The system starts with the user inputting a plaintext message within a Jupyter Notebook cell.
· This plaintext is treated as a sequence of symbols, each with a defined probability, ensuring randomness in the data structure.
· The message can be manually entered or programmatically generated based on specific use cases.
· The entropy of the plaintext is calculated at this stage to assess its randomness and predictability.

2. Encryption Process (Cipher System)

· Once the plaintext is provided, the system applies an encryption algorithm based on Shannon's principles of confusion and diffusion to enhance security.
· A secret key, known only to the sender and the receiver, is used to encrypt the message.
· The encryption function scrambles the original message, converting it into an unintelligible ciphertext.
· The randomness and entropy of the ciphertext are analyzed, ensuring that it maintains a high level of security.

3. Transmission Through a Noisy Channel

· The encrypted message is then transmitted over a simulated noisy channel, mimicking real-world communication systems.
· In an insecure medium, the system assumes the presence of potential attackers who may attempt to intercept the message.
· To counteract such threats, Shannon's theorem ensures that even if an attacker captures the ciphertext, it remains computationally infeasible to retrieve the original plaintext without the decryption key.
· The noisy channel may introduce some distortions, which the system accounts for by implementing error detection mechanisms.

4. Decryption Process at the Receiver’s End

· At the receiving end, the system applies a decryption function using the same secret key to retrieve the original message.

· The decryption process reverses the encryption steps, reconstructing the plaintext from the ciphertext.

· To verify the accuracy and integrity of the recovered message, error correction techniques may be employed.

· If an incorrect key is used, the decrypted output remains unintelligible, reinforcing the security model.

5. Security Analysis and Entropy Measurement

· To assess the robustness of encryption, the system calculates and visualizes Shannon entropy, measuring the uncertainty of the encrypted message.

· A higher entropy value indicates that the encryption is effective, making it harder for attackers to predict or decrypt the message.

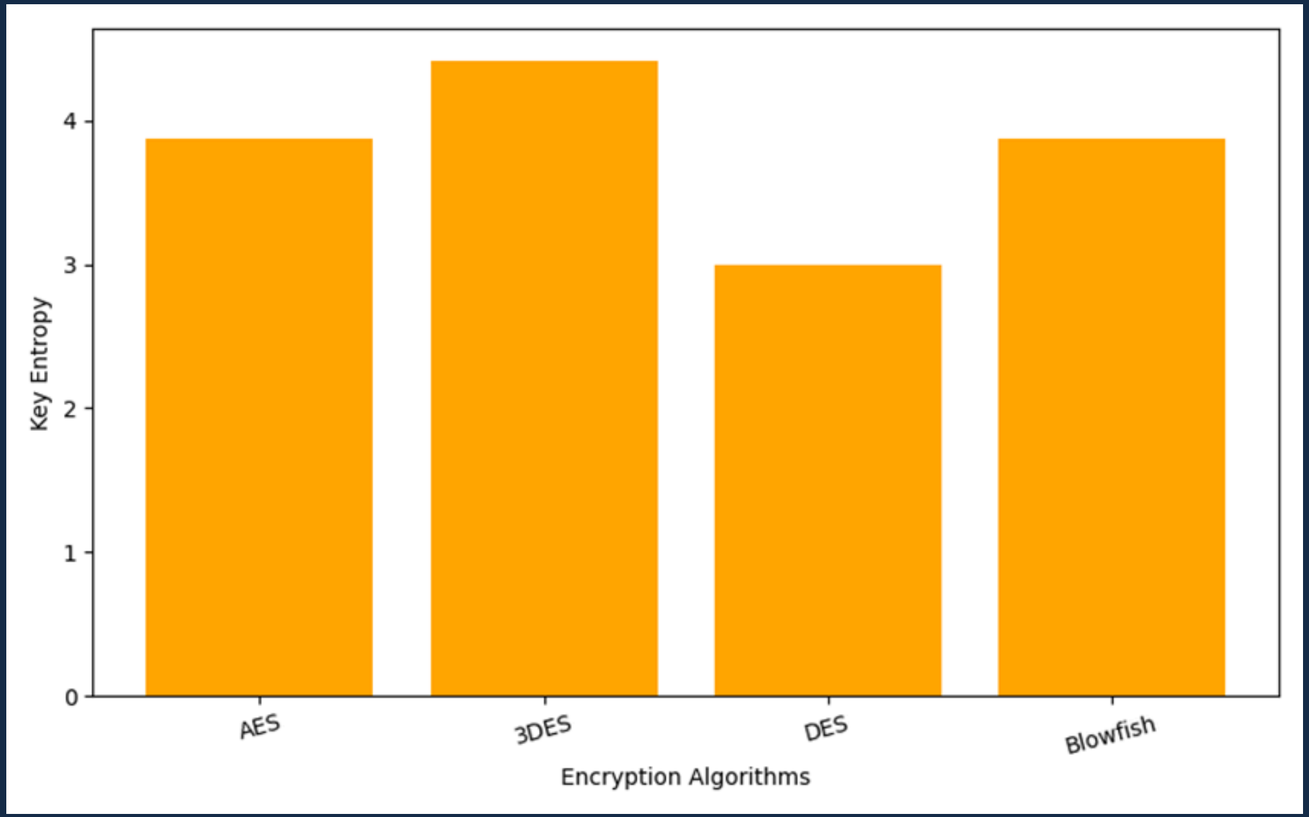
Working Principle

1. Message Generation and Input Processing:
· The user inputs a plaintext message which is the original data intended for secure communication.
· The system preprocesses the message to remove inconsistencies and ensure proper encoding.
2. Key Generation and Encryption:
· The system generates a secret key, ensuring high randomness and unpredictability.
· The plaintext is encrypted using symmetric or asymmetric cryptographic methods.
· The entropy of the generated ciphertext is analyzed to measure security effectiveness.
3. Secure Transmission Through Noisy Channels:
· The encrypted message is transmitted through a simulated noisy channel.
· The system assumes adversarial presence and tests resistance against interception.
4. Decryption and Integrity Verification:
· The receiver decrypts the message using the corresponding secret key.
· The decrypted output is compared with the original plaintext to verify transmission accuracy.
5. Security Evaluation Using Entropy and Probability Distributions:
· The randomness of encryption keys and ciphertext is visualized through entropy graphs and histograms.
· Higher entropy values indicate better security, preventing pattern recognition attacks.
The system performs statistical analysis to identify vulnerabilities in encryption methods

Results and Discussion

1. Entropy Analysis Results:
o The computed entropy values indicate the randomness levels of different encryption algorithms.
o AES and 3DES exhibit higher entropy, ensuring stronger security, while DES shows lower entropy, making it more vulnerable.
2. Key Randomness and Probability Distributions:
o Histogram analysis confirms that secure key generation leads to an even probability distribution.
o Weak keys show patterns that could make encryption susceptible to statistical attacks.
3. Performance Evaluation:
o Encryption and decryption speed are analyzed for different algorithms.
o Results highlight the trade-off between security strength and computational efficiency.
4. Comparison with Traditional Cryptographic Methods:
o The system’s implementation is compared with standard cryptographic practices to assess improvements in security and efficiency.

Symmetric Algorithms

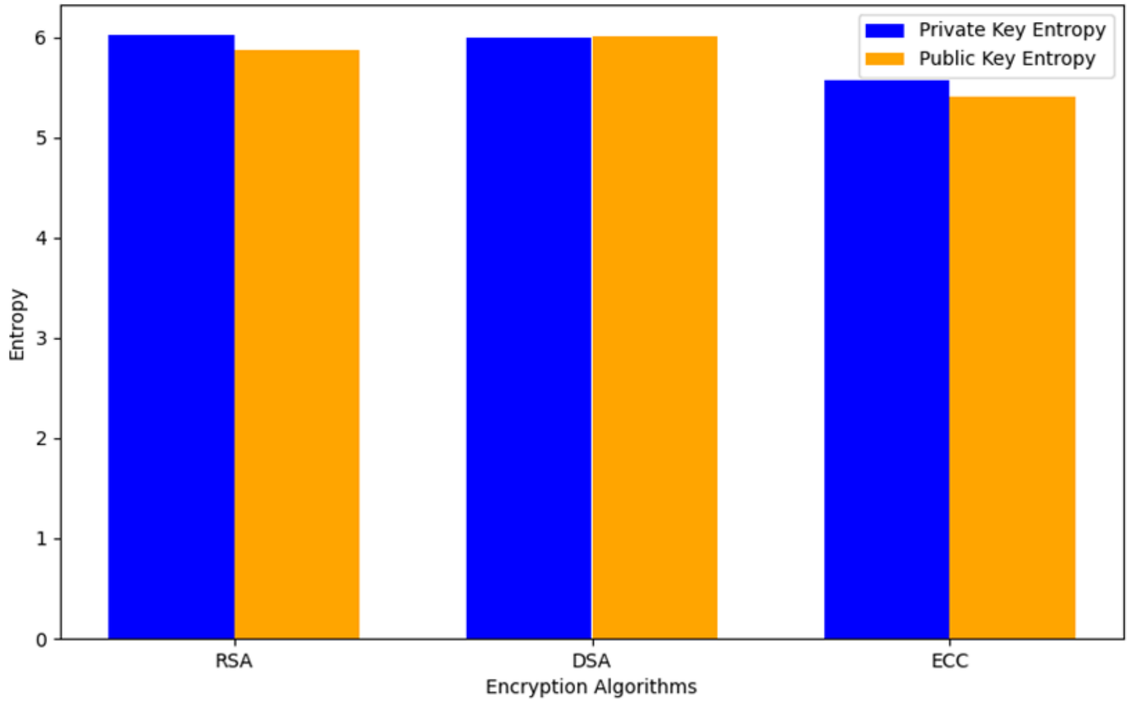


Algorithm	*DES*	*3DES*	*AES*	*BOWLFISH*
Key	3.000	4.303	4.0000	3.8750
Entropy	0 bits	5 bits	bits	bits

TABLE III

KEY ENTROPY TABLE FOR DIFFERENT SYMMETRIC ALGORITHMS

Asymmetric Algorithms



Algorithm	RSA	DSA	ECC
2*Private Key	6.019245 59942364	6.00108 0426728 4	5.615320622 83071 18
2*Public Key	5.879935 20549828	6.01675 0633764 7	5.469152308 482562 302

TABLE IV

KEY ENTROPY TABLE FOR DIFFERENT ASYMMETRIC ALGORITHMS

CONCLUSION

This project presents a secure communication system integrating encryption, decryption, and entropy-based security analysis within a Jupyter Notebook. Using both symmetric and asymmetric encryption, the system ensures data confidentiality and resilience against attacks. Entropy and probability distribution analysis confirm the strength and randomness of the encryption.

Experimental results show that higher entropy values enhance security by reducing vulnerability to cryptanalysis. The system also implements Shannon’s communication model to simulate adversarial scenarios and test encryption robustness. Performance comparisons reveal trade-offs between security and computational efficiency.

The study underscores the value of information-theoretic security in cryptography. This framework lays the groundwork for future research, including algorithm optimization, machine learning integration, and real-world secure messaging applications.

THANK YOU