Cryptographic Algorithms and Implementation Report

I. Introduction

This report outlines the core cryptographic algorithms implemented in a secure communication and encryption platform. It highlights classical and modern encryption techniques, providing a detailed explanation of the underlying logic, implementation steps, and cryptographic processes. These algorithms are crucial for maintaining data confidentiality, integrity, and secure key exchange.

II. Cryptographic Algorithms

1. Data Encryption Standard (DES)

Overview:
DES is a symmetric-key algorithm that encrypts 64-bit blocks of plaintext using a 56-bit key through 16 Feistel network rounds.

Process:

- Input: 64-bit plaintext and a 56-bit key.
- Operations per round:
  - Expansion of 32-bit half-block to 48 bits.
  - XOR with a round key.
  - Substitution via S-boxes.
  - Permutation and swapping halves.

Key Functions:

- ConvertHexToBinary(), ConvertBinaryToHex()
- Permutation(), ShiftLeft()
- XOR(), Encryption(), Decryption()

Encryption Flow:

Hex to Binary conversion.

Initial permutation.

Splitting into left/right halves.

16 rounds of Feistel operations.

Final permutation → Ciphertext.

Decryption:

Similar to encryption but applies subkeys in reverse order (SK16 to SK1).

2. Advanced Encryption Standard (AES)

Overview:
AES is a symmetric encryption algorithm standardized by NIST in 2001. It is stronger and more efficient than DES. AES uses a 128-bit key and performs 10 rounds of encryption.

Key Components:

Key Expansion: Generates round keys using:

Circular byte shift

Byte substitution (S-box)

XOR with round constants

Encryption Steps:

Initial AddRoundKey

9 Rounds:

SubBytes

ShiftRows

MixColumns

AddRoundKey

Final Round (no MixColumns)

Decryption:

Reverses encryption using inverse operations and keys in reverse order.

Mathematics:

Operations performed in Galois Field GF(2^8).

Rijndael S-box for byte substitution.

3. RSA (Rivest–Shamir–Adleman)

Overview:
RSA is an asymmetric public-key cryptosystem used for secure data transmission. It relies on the difficulty of prime factorization.

Steps:

Generate primes p, q.

Compute $n = p * q$ and $\phi(n) = (p-1)*(q-1)$.

Select public key e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1$.

Compute private key d such that $(d * e) \% \phi(n) = 1$.

Encryption:

Convert message to ASCII → encrypt with $C = M^e \bmod n$.

Decryption:

Decrypt with $M = C^d \bmod n$ → Convert ASCII to characters.

Notes:

Modular exponentiation via square-and-multiply.

Developed using PHP.

4. Diffie-Hellman Key Exchange

Overview:
A key exchange protocol enabling two parties to securely share a symmetric key over an insecure channel.

Process:

Generate prime number q and primitive root a.

Each party selects private keys Xa, Xb.

Compute public keys $Ya = a^{Xa} \bmod q$, $Yb = a^{Xb} \bmod q$.

Each computes shared key:

Party A: $K = Yb^{Xa} \bmod q$

Party B: $K = Ya^{Xb} \bmod q$

Functions Used:

findRandomPrime(), findPrimitives(), mpmod() (modular exponentiation)

Notes:

Shared key used in symmetric encryption (e.g., AES).

Developed in JavaScript.

5. El-Gamal Encryption

Overview:
El-Gamal is an asymmetric encryption method built on the Diffie-Hellman key exchange principle and the discrete logarithm problem.

Key Generation:

- Select prime q, primitive root a, private key Xa.
- Compute public key: Ya = a^Xa mod q.

Encryption:

- Generate random k.
- Compute:
  - C1 = a^k mod q
  - K = Ya^k mod q
  - C2 = K * M mod q

Decryption:

- Compute K = C1^Xa mod q.
- Retrieve message: M = C2 * K^-1 mod q.

Notes:

- ASCII-based encoding/decoding of messages.
- Developed in JavaScript.

III. Tools, Languages, and Technologies

Category	Tools & Technologies Used
Development	Visual Studio Code, PHP, JavaScript
Web Frontend	HTML, CSS, Bootstrap, AJAX, jQuery
Backend	PHP, MySQL
Hosting	000webhost
Database	MySQL (for storing keys/messages)

IV. Conclusion

This project integrates classical and modern cryptographic techniques to form a versatile encryption platform. Each algorithm is carefully implemented with mathematical accuracy and integrated into a web or local application environment using various development tools. The system allows secure communication, key sharing, and encryption, making it suitable for educational, research, or lightweight secure messaging platforms.