

BỘ CÔNG THƯƠNG  
TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP HỒ CHÍ MINH  
KHOA CÔNG NGHỆ THÔNG TIN



----oo----

**KHÓA LUẬN CHUYÊN NGÀNH**

**ĐỀ TÀI: TÌM HIỂU THUẬT TOÁN MÃ HÓA DES  
VÀ XÂY DỰNG ỨNG DỤNG MINH HỌA CHI  
TIẾT QUÁ TRÌNH MÃ HÓA VÀ GIẢI MÃ**

TP. HỒ CHÍ MINH, THÁNG 04 NĂM 2025

**BỘ CÔNG THƯƠNG**  
**TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP HỒ CHÍ MINH**  
**KHOA CÔNG NGHỆ THÔNG TIN**



----oo----

**HỌC PHẦN: KHÓA LUẬN KỸ SỰ**

**ĐỀ TÀI: TÌM HIỂU THUẬT TOÁN MÃ HÓA DES VÀ XÂY  
DỤNG ỨNG DỤNG MINH HỌA CHI TIẾT QUÁ TRÌNH MÃ HÓA  
VÀ GIẢI MÃ**

**GVHD: Phạm Tuấn Khiêm**

**Nhóm sinh viên thực hiện đề tài:**

Trưởng nhóm: 2033216386\_ Lý Tiến Đạt

Thành viên:

2033216386 – Lý Tiến Đạt

2033216341 – Lê Quốc Anh

TP. HỒ CHÍ MINH, THÁNG 04 NĂM 2025

## **LỜI CAM ĐOAN**

Nhóm chúng em xin cam đoan đây là công trình nghiên cứu khóa luận chuyên ngành an toàn thông tin của riêng nhóm chúng em và được sự hướng dẫn khoa học của giảng viên ưu tú thầy Phạm Tuân Khiêm. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu cho việc phân tích, nhận xét, đánh giá được chính tác giả là chúng em thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Chúng em xin cam đoan rằng mọi sự giúp đỡ cho việc thực hiện “Bài báo cáo khóa luận” đã được cảm ơn và các thông tin được trích dẫn trong báo cáo đã được chỉ rõ nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào chúng em xin hoàn toàn chịu trách nhiệm về nội dung của chính mình.

**Nhóm sinh viên thực hiện Báo cáo**

*(Đã ký và và ghi rõ họ tên)*

## LỜI CẢM ƠN

Qua một thời gian nghiên cứu, phân công công việc và tiến hành thực hiện đến nay, đề tài “Tìm hiểu thuật toán mã hóa DES và xây dựng minh họa chi tiết quá trình mã hóa và giải mã” đã hoàn thành. Chúng em xin chân thành cảm ơn các thầy cô đã trang bị kiến thức quý báu cho chúng em trong suốt quá trình học tại trường. Đặc biệt là các thầy các cô trong khoa Công nghệ thông tin đã tận tình giảng dạy, chỉ bảo, trang bị cho chúng em những kiến thức cần thiết nhất trong suốt quá trình học tập và nghiên cứu tại khoa, đã tạo mọi điều kiện thuận lợi giúp chúng em.

Đặc biệt không thể thiếu chính là lời cảm ơn chân thành tới giảng viên Phạm Tuấn Khiêm đã tận tình giúp đỡ và hướng dẫn nhóm chúng em hoàn thành tốt đề tài này và cũng không thể thiếu được chính là sự nỗ lực của các thành viên trong nhóm.

Trong quá trình thực hiện đề tài, chúng em không tránh khỏi những thiếu xót cũng như những hạn chế nhất định. Chúng em mong nhận được sự đóng góp ý kiến, chỉ bảo tận tình của thầy cô để đề tài của chúng em hoàn thiện hơn và thiết thực hơn.

## **DANH MỤC VIẾT TẮT**

<b><i>STT</i></b>	<b><i>Ký hiệu chữ viết tắt</i></b>	<b><i>Chữ viết đầy đủ</i></b>
1	DES	Data Encryption Standard
2	DH	Diffie-Hellman
3	NIST	National Institute of Standards
4	ASP.NET	Active Server Pages.NET
5	NSA	National Security Agency
6	IP	Initial Permutation
7	FP	Final Permutation
8	RSA	Rivest Shamir Adleman
9	MITM	Man-in-the-Middle
10	MD5	Message-Digest Algorithm 5

## MỤC LỤC

LỜI CAM ĐOAN .....	2
LỜI CẢM ƠN.....	3
DANH MỤC VIẾT TẮT .....	4
DANH MỤC HÌNH ẢNH .....	8
DANH MỤC BẢNG .....	10
CHƯƠNG 1: PHẦN MỞ ĐẦU.....	11
1.1. GIỚI THIỆU VỀ ĐỀ TÀI .....	11
1.2. LÝ DO CHỌN ĐỀ TÀI .....	12
1.3. MỤC TIÊU CỦA ĐỀ TÀI .....	13
1.3.1. Tìm hiểu chi tiết về ASP.NET và thuật toán DES: .....	13
1.3.2. Tích hợp Diffie-Hellman và hàm băm MD5 trong quá trình sinh khóa cho thuật toán Des: .....	13
1.3.3. Xây dựng một ứng dụng web minh họa quá trình mã hóa và giải mã sử dụng DES trên nền tảng ASP.NET .....	14
1.4. LỊCH SỬ RA ĐỜI CỦA ASP.NET VÀ DES .....	15
CHƯƠNG 2: TỔNG QUAN VỀ MÃ HÓA TRONG MẬT MÃ HỌC... .....	17
2.1. KHÁI NIỆM VỀ MÃ HÓA .....	17
2.2. PHÂN LOẠI MÃ HÓA .....	18
2.2.1. Mã hóa đối xứng .....	18
2.2.2. Mã hóa bất đối xứng.....	18
2.2.3. Kết hợp mã hóa đối xứng và mã hóa bất đối xứng .....	19
2.3. LỢI ÍCH CỦA MÃ HÓA .....	19
2.3.1. Hệ thống bảo mật của doanh nghiệp .....	19
2.3.2. Bảo vệ dữ liệu và đảm bảo tính toàn vẹn. ....	20
2.3.3. Bảo mật trên nhiều thiết bị và linh hoạt .....	21
CHƯƠNG 3: THUẬT TOÁN MÃ HÓA DES .....	22
3.1. KHÁI NIỆM VÀ CẤU TRÚC THUẬT TOÁN MÃ HÓA DES .	22
3.1.1. Giới thiệu chung về DES .....	22

3.1.2. Cấu trúc và nguyên lí hoạt động của DES .....	23
<b>3.2. CÁC THÀNH PHẦN BẢO MẬT TRONG DES .....</b>	<b>24</b>
3.2.1. Hàm f .....	24
3.2.2. Khóa con (Subkeys) .....	28
3.2.3. Hoán vị đầu và hoán vị cuối (IP&FP).....	30
<b>3.3. QUY TRÌNH MÃ HÓA VÀ GIẢI MÃ TRONG DES .....</b>	<b>31</b>
3.3.1. Khởi tạo và phân bố khóa.....	31
3.3.2. Quy trình mã hóa trong DES .....	37
3.3.3. Quy trình giải mã trong DES: .....	39
3.3.4. Ví dụ thực tiễn:.....	40
<b>3.4. ƯU NHƯỢC ĐIỂM VÀ CÁC YẾU TỐ KHÁC CỦA DES .....</b>	<b>57</b>
3.4.1. Ưu điểm của DES.....	57
3.4.2. Nhược điểm của DES .....	57
<b>3.5. TÍCH HỢP THUẬT TOÁN DIFFIE-HELLMAN, RSA VÀ MD5 ĐỂ TĂNG CƯỜNG BẢO MẬT CHO HỆ THỐNG MÃ HÓA DES .....</b>	<b>58</b>
3.5.1. Lý do sử dụng các thuật toán hỗ trợ trong hệ thống DES:.....	58
3.5.2. Mô hình trao đổi khóa của Diffie-Hellman: .....	59
3.5.3. Tăng cường xác minh danh tính bằng thuật toán RSA .....	61
3.5.4. Chuẩn hóa khóa bằng hàm băm MD5.....	62
3.5.5. Mô hình tích hợp ba lớp bảo mật .....	63
<b>CHƯƠNG 4: XÂY DỰNG ỨNG DỤNG WEB MINH HỌA THUẬT TOÁN DES .....</b>	<b>65</b>
<b>4.1. THIẾT KẾ GIAO DIỆN ỨNG DỤNG WEB .....</b>	<b>65</b>
4.1.1. Khối giới thiệu trang chủ:.....	65
4.1.2. Khối tính năng nổi bật và kết hợp với các thuật toán khác:... 65	65
4.1.3. Trang mô tả thuật toán và hướng dẫn sử dụng: .....	67
4.1.4. Thông tin nhóm phát triển trang web:.....	69
4.1.5. Phản hồi của người dùng: .....	70
4.1.6. Footer .....	71

4.2. MÔ TẢ CHỨC NĂNG CỦA ỨNG DỤNG .....	72
4.2.1. Tạo khóa chung bằng thuật toán Diffie-Hellman .....	72
4.2.2. Tạo cặp khóa RSA cho chữ ký số .....	74
4.2.3. Xác thực khóa bằng chữ ký số RSA .....	76
4.2.4. Băm khóa bằng MD5 và trích xuất khóa DES .....	78
4.2.5. Mã hóa dữ liệu với DES .....	81
4.2.6. Giải mã dữ liệu với DES .....	82
4.2.7. Tổng kết toàn bộ quy trình minh họa:.....	84
4.3. TRIỂN KHAI VÀ XUẤT BẢN WEB LÊN INTERNET .....	85
KẾT LUẬN .....	93
TÀI LIỆU THAM KHẢO.....	94

## **DANH MỤC HÌNH ẢNH**

Hình 1.1. Nền tảng ASP.NET của Microsoft.....	15
Hình 1.2. Mô hình mã hóa quy ước.....	16
Hình 2.1. Mã hóa dữ liệu .....	17
Hình 2.2. Mã hóa đối xứng .....	18
Hình 2.3. Mã hóa bất đối xứng.....	19
Hình 3.1. Mã hóa đối xứng DES.....	22
Hình 3.2. Sơ đồ tính hàm mã hóa f(R,k).....	24
Hình 3.3. Sơ đồ quá trình sinh khóa con (1).....	29
Hình 3.4: Sơ đồ tạo khóa con (2) .....	32
Hình 3.5. Biểu diễn chuỗi 64 bit x thành 2 phần L và R.....	37
Hình 3.6. Một vòng ( vòng thứ i ) của DES.....	39
Hình 3.7. Sơ đồ mã hóa DES .....	40
Hình 3.8. Mô hình ví dụ trao đổi màu sơn giữa Alice và Bob .....	61
Hình 4.1. Giao diện trang chủ khi truy cập website .....	65
Hình 4.2. Giao diện hiển thị tính năng hệ thống .....	66
Hình 4.3. Giao diện mô tả thuật toán DES 1 .....	67
Hình 4.4. Giao diện mô tả thuật toán DES 2 .....	68
Hình 4.5. Giao diện phần đầu hướng dẫn sử dụng web .....	69
Hình 4.6. Giao diện phần cuối hướng dẫn sử dụng web .....	69
Hình 4.7. Thông tin nhà phát triển và liên hệ .....	70
Hình 4.8. Phản hồi của người dùng 1 .....	70
Hình 4.9. Phản hồi của người dùng 2 .....	71
Hình 4.10. Footer chính của website .....	71
Hình 4.11. Giao diện nhập tham số Diffie-Hellman trong hệ thống .....	73
Hình 4.12. Giao diện kết quả Diffie-Hellman trong hệ thống .....	74
Hình 4.13. Giao diện tạo cặp khóa RSA.....	76
Hình 4.14. Giao diện ký số RSA .....	77

Hình 4.15. Giao diện xác minh danh tính RSA .....	78
Hình 4.16. Băm khóa với MD5 .....	79
Hình 4.17. Kết quả băm khóa với MD5 .....	79
Hình 4.18. Trích xuất từ MD5.....	80
Hình 4.19. Chuyển đổi định dạng khóa .....	80
Hình 4.20. Mã hóa thuật toán DES.....	82
Hình 4.21. Giải mã thuật toán DES .....	83
Hình 4.22. Trang đăng nhập của Somee.....	85
Hình 4.23. Đặt tên domain cho website.....	86
Hình 4.24. Giao diện khi tạo xong website.....	86
Hình 4.25. Thông tin FTP và link website.....	87
Hình 4.26. Giao diện khi nháp publish .....	87
Hình 4.27. Publish folder .....	88
Hình 4.28. Đường dẫn lưu trữ .....	88
Hình 4.29. Quá trình publish đang thực hiện .....	89
Hình 4.30. Quá trình publish thành công.....	89
Hình 4.31. Giao diện login vào WinSCP .....	90
Hình 4.32. Giao diện chờ kết nối WinSCP .....	90
Hình 4.33. Giao diện thao tác sau khi kết nối .....	91
Hình 4.34. Quá trình upload file code .....	91
Hình 4.35. Upload file code xong Somee thành công.....	92
Hình 4.36. Giao diện website đã được public lên Internet .....	92

## **DANH MỤC BẢNG**

Bảng 1.1. Mô hình quá trình mã hóa quy ước .....	16
Bảng 3.1. 8 Sbox của thuật toán DES.....	27
Bảng 3.2. Bảng P .....	27
Bảng 3.3. Bảng mở rộng E .....	28
Bảng 3.4. Bảng dịch bit .....	29
Bảng 3.5. Bảng hoán vị IP và IP-1 .....	30
Bảng 3.6. Bảng hoán vị PC-1 .....	33
Bảng 3.7. Bảng hoán vị PC-2 .....	36
Bảng 3.8. Quy trình mã hóa DES .....	38
Bảng 3.9. Bảng quy trình trao đổi khóa Diffie-Hellman .....	60
Bảng 4.1. Ví dụ các tham số Diffie-Hellman .....	72
Bảng 4.2. Ví dụ cách tích khóa công khai Diffie-Hellman .....	73
Bảng 4.3. Ví dụ cách tích khóa chung Diffie-Hellman .....	74
Bảng 4.4. Ví dụ tạo cặp khóa RSA .....	75
Bảng 4.5. Ví dụ các tham số tạo chữ ký .....	77
Bảng 4.6. Ví dụ băm khóa và chuyển đổi .....	81
Bảng 4.7. Các dạng dữ liệu đầu vào .....	81
Bảng 4.8. Tổng kết giải mã .....	83
Bảng 4.9. Toàn bộ quy trình .....	84

# CHƯƠNG 1: PHẦN MỞ ĐẦU

## 1.1. GIỚI THIỆU VỀ ĐỀ TÀI

Sự tăng trưởng nhanh chóng của thế giới đã tác động đến cả công việc kinh doanh và tiêu dùng với sự hứa hẹn về việc thay đổi cách mà con người làm việc và sống. Nhưng sự bát an lớn nhất được nhắc đến là việc an toàn thông tin trên không gian mạng, đặc biệt là khi những thông tin có tính nhạy cảm được gửi đi trên môi trường mạng. Mã hóa trong máy tính được thực hiện dựa trên nền tảng của khoa học mật mã, một lĩnh vực đã được con người sử dụng từ thời xa xưa. Trước khi bước vào kỷ nguyên số, việc sử dụng mật mã chủ yếu nằm trong tay các chính phủ, đặc biệt là phục vụ các hoạt động quân sự. Hiện nay, phần lớn các phương pháp mã hóa đều dựa vào máy tính, bởi các loại mã được con người tạo thủ công rất dễ bị phá vỡ bằng các công cụ tính toán hiện đại.

Khoa học mật mã không chỉ đơn giản phục vụ cho việc mã hóa và giải mã thông tin, mà còn liên quan đến nhiều vấn đề khác cần được nghiên cứu và xử lý như chứng thực nguồn gốc nội dung thông tin (chữ ký số), chứng nhận xác thực về người sở hữu mã hóa, chứng nhận của khóa công khai, các thủ tục đảm bảo an toàn trong việc trao đổi dữ liệu và tiến hành giao dịch điện tử đảm bảo an toàn qua môi trường mạng.

Bảo mật thông tin luôn đóng vai trò vô cùng quan trọng, đặc biệt trong việc bảo vệ các thông điệp nhạy cảm khỏi các mối đe dọa bên ngoài. Từ thời cổ đại đến kỷ nguyên số hiện nay, nhu cầu về các phương pháp mã hóa và giải mã thông tin để bảo vệ dữ liệu đã trở nên cấp thiết.

## 1.2. LÝ DO CHỌN ĐỀ TÀI

Thuật toán DES (Data Encryption Standard) là một trong những thuật toán mã hóa đối xứng đầu tiên được chuẩn hóa và sử dụng rộng rãi trong lĩnh vực bảo mật thông tin. Dù ngày nay đã có nhiều thuật toán mạnh hơn như AES thay thế, DES vẫn đóng vai trò nền tảng trong lịch sử phát triển của mật mã học hiện đại, đồng thời là ví dụ điển hình giúp người học hiểu rõ nguyên lý hoạt động của mã hóa đối xứng.

Việc nghiên cứu thuật toán DES giúp người học làm quen với các khái niệm cốt lõi trong khoa học mật mã như hoán vị (permutation), thay thế (substitution), khóa đối xứng, và quy trình mã hóa – giải mã thông tin. DES sử dụng cùng một khóa cho cả hai quá trình mã hóa và giải mã, giúp đơn giản hóa hệ thống nhưng cũng đặt ra thách thức về phân phối và quản lý khóa, từ đó mở ra cơ hội tìm hiểu sâu hơn về các vấn đề thực tế trong bảo mật.

Mặc dù không còn được khuyến khích sử dụng trong các hệ thống bảo mật hiện đại do độ dài khóa ngắn (56 bit), DES vẫn là một công cụ giáo dục hiệu quả để minh họa cách các thuật toán mã hóa hoạt động ở cấp độ bit, qua các vòng lặp xử lý, hàm Feistel, và các bảng thay thế (S-box). Bằng cách lập trình mô phỏng thuật toán DES, người học có thể hiểu rõ hơn quá trình biến đổi dữ liệu từ văn bản thuần sang dạng mã hóa và ngược lại.

Hơn nữa, trong bối cảnh an ninh mạng ngày càng phức tạp, việc mã hóa thông tin vẫn luôn là lớp bảo vệ đầu tiên và thiết yếu. Dù các phần mềm phòng chống mã độc có thể phát hiện các mối đe dọa, vẫn luôn tồn tại nguy cơ bị xâm nhập trái phép. Khi đó, việc mã hóa thông tin bằng các

thuật toán như DES giúp đảm bảo rằng dữ liệu, dù có bị đánh cắp, cũng không thể bị đọc nếu không có khóa giải mã phù hợp.

Với tính chất đơn giản, rõ ràng và cấu trúc học thuật chặt chẽ, thuật toán DES là lựa chọn lý tưởng để triển khai ứng dụng minh họa quá trình mã hóa – giải mã thông tin. Từ đó, đề tài khóa luận này của chúng em giúp người học tiếp cận mật mã học hiệu quả hơn với một cách trực quan và nền tảng vững chắc, phục vụ cho việc học tập và nghiên cứu các giải pháp bảo mật hiện đại trong lĩnh vực an ninh mạng.

### **1.3. MỤC TIÊU CỦA ĐỀ TÀI**

#### **1.3.1. Tìm hiểu chi tiết về ASP.NET và thuật toán DES:**

Nghiên cứu kiến trúc, đặc điểm, tính năng và cách triển khai mô hình hoạt động của một ứng dụng web bằng ASP.NET.

Tìm hiểu chi tiết về thuật toán mã hóa DES: cấu trúc 16 vòng mã hóa, nguyên lý hoạt động, quy trình tạo 16 khóa con, các bước mã hóa và giải mã, các thành phần như S-box, P-box, Permutation....

Phân tích các ưu nhược điểm, những hạn chế và phạm vi ứng dụng thực tế của DES trong bảo mật thông tin.

#### **1.3.2. Tích hợp Diffie-Hellman và hàm băm MD5 trong quá trình sinh khóa cho thuật toán Des:**

Tìm hiểu nguyên lý hoạt động của thuật toán Diffie-Hellman trong trao đổi khóa an toàn.

Triển khai cơ chế sinh khóa chung bằng Diffie-Hellman để mô phỏng quá trình chia sẻ khóa bí mật giữa hai phía.

Băm khóa chung bằng thuật toán MD5 để tạo chuỗi HEX ngẫu nhiên, sau đó trích xuất 16 ký tự đầu tiên, chuyển sang dạng nhị phân để sử dụng làm khóa cho thuật toán DES.

Mô phỏng quy trình tích hợp này trong ứng dụng ASP.NET để tăng tính thực tiễn và nâng cao sự bảo mật.

### **1.3.3. Xây dựng một ứng dụng web minh họa quá trình mã hóa và giải mã sử dụng DES trên nền tảng ASP.NET**

Thiết kế giao diện web thân thiện và triển khai một ứng dụng cho phép người dùng nhập văn bản và khóa mã hóa, thực hiện mã hóa và giải mã bằng thuật toán DES.

Triển khai thuật toán DES bằng ngôn ngữ C# trong ASP.NET, bao gồm các bước: mở rộng khóa, tạo 16 khóa con, xử lý các bảng chuyển vị, vòng lặp mã hóa/giải mã.

Kết quả của thuật toán được hiển thị chi tiết từng bước xử lý trong quá trình mã hóa – giải mã của DES nhằm hiểu rõ hơn và minh họa trực quan thuật toán.

Hiển thị song song các bảng thực hiện trong quá trình mã hóa giải mã của thuật toán để có thể trải nghiệm tốt hơn.

Trang mô tả trình bày bảng sinh khóa con, biểu diễn luồng dữ liệu giữa các vòng để giúp người học hình dung rõ cơ chế hoạt động của DES.

Tích hợp phần sinh khóa từ Diffie-Hellman vào quy trình tổng thể để hoàn thiện mô hình truyền thông bảo mật.

## **1.4. LỊCH SỬ RA ĐỜI CỦA ASP.NET VÀ DES**

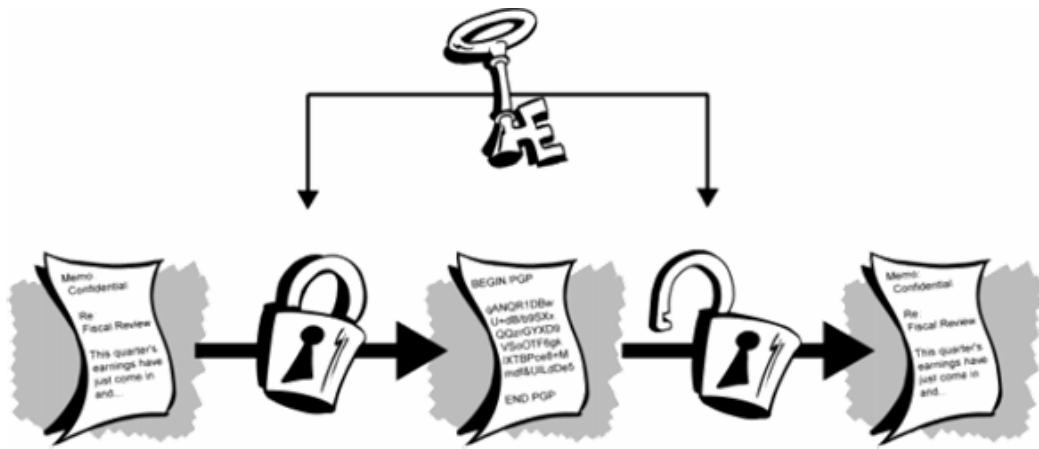
ASP.NET là một phần của nền tảng .NET do Microsoft phát triển, lần đầu tiên được giới thiệu vào tháng 1 năm 2002 như một sự kế thừa của công nghệ ASP (Active Server Pages). ASP.NET được thiết kế nhằm tạo điều kiện thuận lợi cho việc xây dựng các ứng dụng web động, bảo mật và có khả năng mở rộng cao. Qua nhiều phiên bản, ASP.NET đã phát triển từ Web Forms đến MVC và hiện nay là ASP.NET Core – nền tảng đa hệ điều hành, hiệu suất cao và mã nguồn mở. ASP.NET hiện là một công cụ phổ biến trong phát triển web doanh nghiệp.



**Hình 1.1.** Nền tảng ASP.NET của Microsoft

Thuật toán DES được phát triển bởi IBM vào đầu những năm 1970, dưới sự lãnh đạo của nhà mật mã học Horst Feistel. Năm 1977, DES chính thức được chuẩn hóa bởi Viện Tiêu chuẩn và Công nghệ Quốc gia Hoa Kỳ (NIST) và trở thành một chuẩn mã hóa chính thức của chính phủ Mỹ. DES từng là nền tảng bảo mật dữ liệu trong các ứng dụng thương mại và chính phủ trong suốt nhiều năm. Tuy nhiên, do độ dài khóa chỉ 56 bit, DES đã dần trở nên không an toàn trước các cuộc tấn công brute-force và hiện nay đã được thay thế bởi các thuật toán mạnh hơn như Triple DES.

và AES. Tuy vậy, DES vẫn là một ví dụ kinh điển để nghiên cứu về thiết kế thuật toán mã hóa đối xứng.



**Hình 1.2.** Mô hình mã hóa quy ước

Thông điệp	Khóa	Mã hóa	Thông điệp	Khóa	Giải mã	Thông điệp
Nguồn			Đã mã hóa			Đã giải mã

**Bảng 1.1.** Mô hình quá trình mã hóa quy ước

## CHƯƠNG 2: TỔNG QUAN VỀ MÃ HÓA TRONG MẶT MÃ HỌC

### 2.1. KHÁI NIỆM VỀ MÃ HÓA

Mã hóa là cách xáo trộn dữ liệu chỉ để hai bên trao đổi thông tin có thể hiểu được. Về mặt kỹ thuật, đó là quá trình chuyển đổi văn bản gốc sang bản mã. Nói một cách đơn giản hơn, mã hóa lấy dữ liệu có thể đọc được và thay đổi nó để dữ liệu này không giống như ban đầu. Mã hóa yêu cầu sử dụng khóa mã hóa: một tập hợp các giá trị toán học mà cả người gửi và người nhận tin nhắn được mã hóa đều biết. Mặc dù dữ liệu được mã hóa xuất hiện ngẫu nhiên, mã hóa tiến hành theo cách hợp lý, có thể dự đoán được, để bên nhận sử dụng khóa để mã hóa dữ liệu, biến nó trở lại thành bản dữ liệu ban đầu.



**Hình 2.1.** Mã hóa dữ liệu

Mã hóa an toàn thực sự sẽ đủ phức tạp để bên thứ ba không thể giải mã được bằng brute force. Dữ liệu có thể được mã hóa “ở trạng thái nghỉ”, khi nó được lưu trữ hoặc “quá cảnh” trong khi nó đang được truyền đi nơi khác.

## 2.2. PHÂN LOẠI MÃ HÓA

### 2.2.1. Mã hóa đối xứng

Trong hệ thống mã hóa đối xứng, quá trình mã hóa và giải mã một thông điệp sử dụng cùng một mã khóa gọi là khóa bí mật (secret key) hay khóa đối xứng (symmetric key). Khóa mã hóa và khóa giải mã có mối liên hệ chặt chẽ, có thể là trùng nhau hoàn toàn hoặc chỉ khác biệt thông qua một phép biến đổi dễ hiểu giữa hai khóa. Do đó, vấn đề bảo mật thông tin đã mã hóa hoàn toàn phụ thuộc vào việc giữ bí mật nội dung của mã khóa đã được sử dụng.

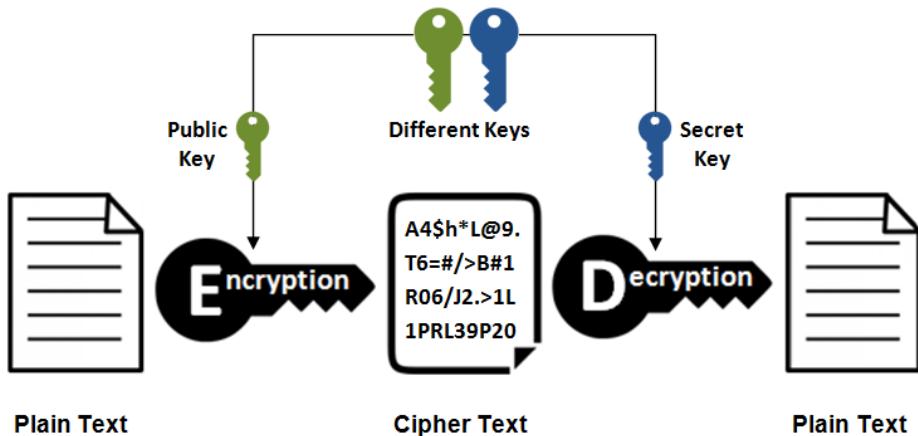


**Hình 2.2.** Mã hóa đối xứng

### 2.2.2. Mã hóa bất đối xứng

Nếu như vấn đề khó khăn đặt ra đối với các phương pháp mã hóa quy ước chính là bài toán trao đổi mã khóa thì ngược lại, các phương pháp mã hóa khóa công cộng giúp cho việc trao đổi mã khóa trở nên dễ dàng hơn. Nội dung của khóa công cộng (public key) không cần phải giữ bí mật như đối với khóa bí mật trong các phương pháp mã hóa quy ước. Sử dụng khóa công cộng, chúng ta có thể thiết lập một quy trình an toàn để truy đổi khóa bí mật được sử dụng trong hệ thống mã hóa quy ước.

## Asymmetric Encryption



**Hình 2.3.** Mã hóa bất đối xứng

### 2.2.3. Kết hợp mã hóa đối xứng và mã hóa bất đối xứng

Các phương pháp mã hóa đối xứng có ưu điểm xử lý rất nhanh và khả năng bảo mật cao so với các phương pháp mã hóa khóa bất đối xứng nhưng lại gặp phải vấn đề khó khăn trong việc trao đổi mã khóa. Ngược lại, các phương pháp mã hóa khóa bất đối xứng tuy xử lý thông tin chậm hơn nhưng lại cho phép người sử dụng trao đổi mã khóa dễ dàng hơn. Do đó, trong các ứng dụng thực tế, chúng ta cần phối hợp được ưu điểm của mỗi phương pháp mã hóa để xây dựng hệ thống mã hóa và bảo mật thông tin hiệu quả và an toàn.

## 2.3. LỢI ÍCH CỦA MÃ HÓA

### 2.3.1. Hệ thống bảo mật của doanh nghiệp

Nhiều doanh nghiệp đã đầu tư vào các hệ thống phát hiện xâm nhập và tường lửa truyền thông, nhưng vẫn tiếp tục tìm kiếm những giải pháp bảo mật hiệu quả hơn để bảo vệ dữ liệu. Trước các mối đe dọa từ cả tin tặc bên ngoài lẫn nguy cơ rò rỉ từ bên trong, việc đảm bảo an toàn dữ liệu ở mọi trạng thái (lưu trữ, truyền tải, sử dụng) là một thách thức lớn đối với

các giải pháp cũ. Do đó, ngày càng nhiều doanh nghiệp lựa chọn triển khai các giải pháp mã hóa như một lớp bảo vệ quan trọng trong hệ thống bảo mật tổng thể. Mã hóa đặc biệt phù hợp với các tổ chức có nhu cầu xử lý và truyền tải lượng lớn dữ liệu, đồng thời yêu cầu tính bảo mật cao. Những lo ngại trước đây về việc mã hóa ảnh hưởng đến hiệu năng hay trải nghiệm người dùng đã dần được khắc phục, khiến mã hóa ngày càng trở nên phổ biến.

Thực tế cho thấy, vi phạm dữ liệu là vấn đề nghiêm trọng đối với các doanh nghiệp ở mọi quy mô. Nhiều công ty nhỏ vẫn làm tưởng rằng mối nguy này chỉ xảy ra với các tập đoàn lớn. Tuy nhiên, mã hóa không chỉ giúp bảo vệ dữ liệu khỏi bị truy cập trái phép mà còn mang lại nhiều lợi ích khác mà doanh nghiệp có thể chưa nhận ra. Việc đánh giá và áp dụng mã hóa trong hệ thống bảo mật là bước đi cần thiết để nâng cao khả năng phòng thủ trước các mối nguy hiện đại.

### **2.3.2. Bảo vệ dữ liệu và đảm bảo tính toàn vẹn.**

Một trong những mối quan ngại phổ biến của các tổ chức là liệu quá trình mã hóa có ảnh hưởng đến tính toàn vẹn của dữ liệu hay không. Trong khi việc đánh cắp dữ liệu là một rủi ro rõ ràng, tin tặc cũng có thể thực hiện các hành vi tấn công nhằm thay đổi hoặc làm giả dữ liệu. Mã hóa giúp bảo vệ dữ liệu khỏi sự thay đổi trái phép và đảm bảo rằng người nhận có thể xác thực tính nguyên vẹn của dữ liệu thông qua các cơ chế kiểm tra.

Đối với các doanh nghiệp đang tìm kiếm giải pháp an toàn sau các sự cố rò rỉ thông tin, mã hóa đóng vai trò then chốt trong việc phục hồi và phòng ngừa. Nó cho phép triển khai bảo mật đồng bộ trên nhiều thiết bị, đảm bảo dữ liệu luôn được mã hóa khi gửi, nhận hoặc chia sẻ. Từ đó, doanh nghiệp có thể yên tâm rằng dữ liệu của mình không bị thay đổi và được kiểm soát chặt chẽ trong suốt vòng đời sử dụng.

### **2.3.3. Bảo mật trên nhiều thiết bị và linh hoạt**

Với sự phổ biến của điện thoại thông minh và thiết bị di động, việc bảo vệ dữ liệu khi được lưu trữ hoặc truyền qua các thiết bị này ngày càng trở nên quan trọng. May mắn thay, các giải pháp mã hóa hiện đại cho phép áp dụng các biện pháp bảo mật tương đương trên mọi thiết bị, từ máy tính cá nhân đến thiết bị di động, giúp giảm thiểu rủi ro thất thoát dữ liệu trong môi trường làm việc linh hoạt hiện nay.

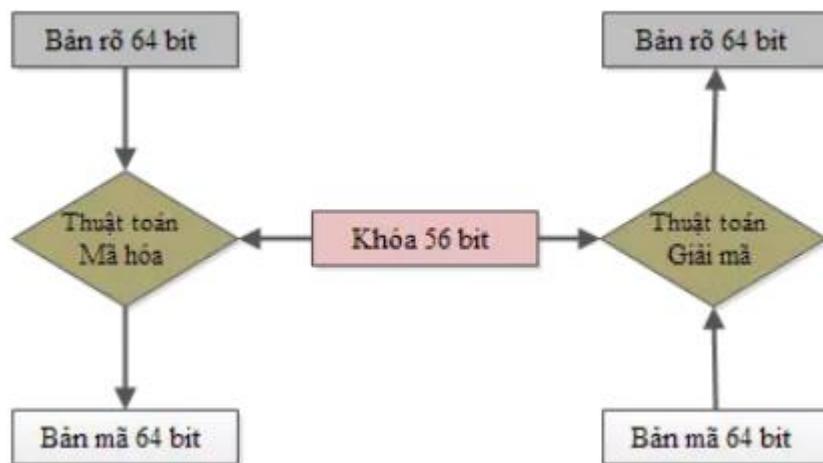
Mã hóa còn giúp kiểm soát hiệu quả các tình huống rủi ro phát sinh từ thiết bị cá nhân, đồng thời kết hợp với cơ chế xác thực để ngăn chặn truy cập trái phép. Bên cạnh đó, dữ liệu trong quá trình vận chuyển — vốn là giai đoạn dễ bị tấn công nhất — cũng được bảo vệ tốt hơn. Mặc dù các giao thức như SSL/TLS đã trở thành tiêu chuẩn, chúng vẫn tồn tại một số điểm yếu. Việc bổ sung các lớp mã hóa bổ sung giúp tăng cường bảo mật trong quá trình truyền tải dữ liệu, đặc biệt khi chia sẻ hoặc lưu trữ trên nền tảng đám mây.

## CHƯƠNG 3: THUẬT TOÁN MÃ HÓA DES

### 3.1. KHÁI NIỆM VÀ CẤU TRÚC THUẬT TOÁN MÃ HÓA DES

#### 3.1.1. Giới thiệu chung về DES

DES (Data Encryption Standard) là một trong những thuật toán mã hóa đối xứng đầu tiên được tiêu chuẩn hóa và sử dụng rộng rãi trong lĩnh vực bảo mật thông tin. Được phát triển vào đầu những năm 1970 bởi IBM dưới tên gọi ban đầu là “Lucifer”, thuật toán này sau đó được chính phủ Hoa Kỳ chọn làm tiêu chuẩn mã hóa chính thức vào năm 1977 thông qua Cơ quan An ninh Quốc gia Hoa Kỳ (NSA) và Viện Tiêu chuẩn và Công nghệ Quốc gia Hoa Kỳ (NIST).



Hình 3.1. Mã hóa đối xứng DES

DES sử dụng một khóa bí mật 64-bit (trong đó chỉ có 56 bit là khóa thực, 8 bit còn lại dùng để kiểm tra chẵn lẻ) để mã hóa và giải mã dữ liệu theo phương pháp mã hóa khối (block cipher). Cụ thể, dữ liệu đầu vào sẽ được chia thành các khối có độ dài 64-bit, mỗi khối này sau đó sẽ trải qua 16 vòng xử lý (16 rounds) bao gồm các bước thay thế, hoán vị, chia tách, kết hợp và sử dụng khóa con riêng biệt cho mỗi vòng.

Tại thời điểm ra đời, DES được xem là một bước tiến quan trọng trong công nghệ mã hóa với khả năng triển khai hiệu quả trên cả phần cứng và phần mềm. Tuy nhiên, cùng với sự phát triển của công nghệ, thuật toán này dần trở nên kém an toàn do độ dài khóa ngắn, dễ bị phá vỡ bởi tấn công brute-force. Tuy vậy, việc nghiên cứu DES vẫn có giá trị cao trong việc giúp người học hiểu rõ cơ chế hoạt động của mã hóa đối xứng.

### **3.1.2. Cấu trúc và nguyên lí hoạt động của DES**

#### **3.1.2.1. Mạng Feistel trong DES**

DES áp dụng cấu trúc mạng Feistel, một phương pháp phổ biến trong thiết kế các thuật toán mã hóa khối. Mạng Feistel chia khối dữ liệu 64 bit thành hai nửa (trái và phải), và qua mỗi vòng mã hóa, nửa bên phải được xử lý qua một hàm F, kết quả sau đó được XOR với nửa bên trái, sau đó hai nửa này đổi chỗ cho vòng tiếp theo.

Trong DES, mỗi vòng mã hóa sử dụng một khóa con (subkey) được tạo ra từ khóa chính 56 bit thông qua một quá trình hoán vị và dịch vòng. Hàm F trong mỗi vòng sử dụng các kỹ thuật như mở rộng bit, hoán vị (P-box), và các hộp thay thế S-box để tăng độ phức tạp và tính không tuyến tính. Thuật toán DES sử dụng tổng cộng 16 vòng mã hóa được thực hiện trên các khối dữ liệu 64 bit.

#### **3.1.2.2. Đặc điểm chính trong cấu trúc DES**

Độ dài khóa cố định: DES sử dụng khóa có độ dài 56 bit hiệu dụng (64 bit đầu vào với 8 bit kiểm tra chẵn lẻ), đây là tiêu chuẩn vào thời điểm được phát triển.

Mã hóa khối 64 bit: DES mã hóa dữ liệu theo từng khối 64 bit, phù hợp với khả năng xử lý của phần cứng thời kỳ đầu.

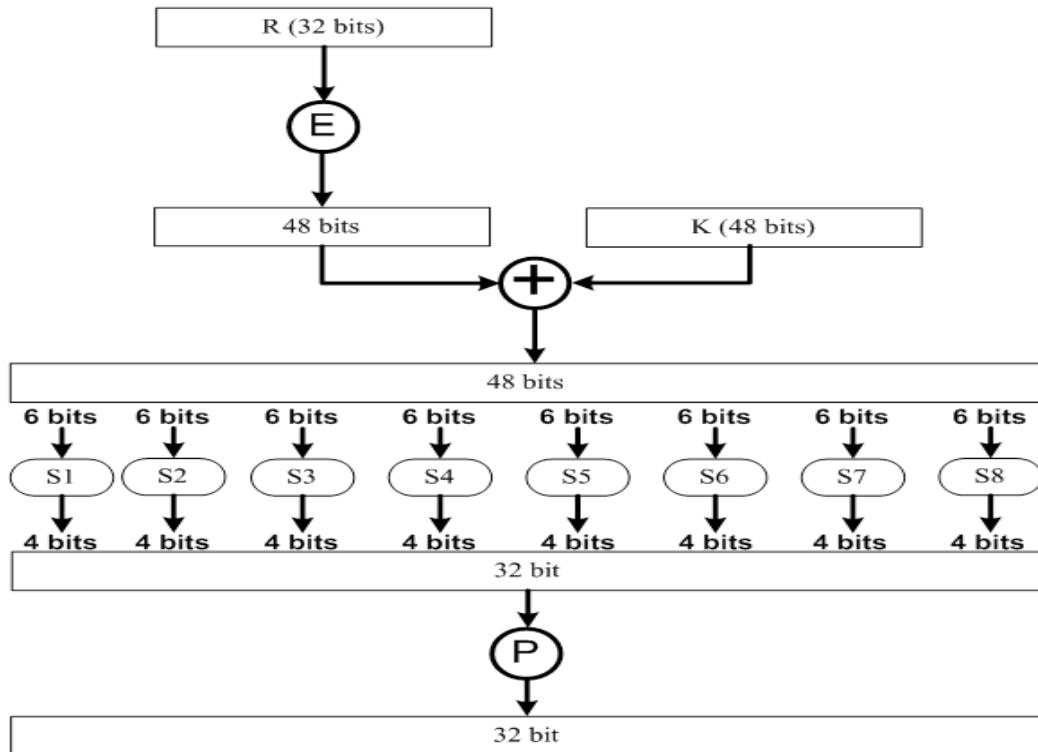
Tính bảo mật nền tảng: DES sử dụng các kỹ thuật mã hóa như cấu trúc mạng Feistel, các bảng thay thế (S-Box), bảng hoán vị (P-Box) và phép XOR để tăng độ phức tạp và đảm bảo tính không tuyến tính trong quá trình mã hóa.

### 3.2. CÁC THÀNH PHẦN BẢO MẬT TRONG DES

#### 3.2.1. Hàm f

Hàm f trong mỗi vòng lặp là trung tâm, lõi xử lý chính của DES:

- Nhận 32 bit đầu vào.
- Mở rộng thành 48 bit bằng bảng E.
- XOR với khóa con 48 bit.
- Chia thành 8 nhóm 6 bit → đưa qua bảng 8 S-box → ra 32 bit.
- Áp dụng hoán vị P trên 32 bit.



**Hình 3.2.** Sơ đồ tính hàm mã hóa f(R,k)

### 3.2.1.1. S-box (Substitution boxes)

Là thành phần tạo nên tính phi tuyến duy nhất trong thuật toán DES. Có tổng cộng 8 S-box, mỗi hộp thay thế một dãy 6 bit thành 4 bit theo bảng quy định sẵn. Các S-box có vai trò then chốt trong việc làm phức tạp hóa mối quan hệ giữa dữ liệu và khóa, từ đó đảm bảo tính ẩn hóa.

Tóm lại: 8 bảng thay thế 6 bit → 4 bit, góp phần phi tuyến hóa để chống tấn công.

Sbox1															
14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Sbox2															
15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

Sbox3															
10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

<b>Sbox4</b>															
7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	3	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

<b>Sbox5</b>															
2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

<b>Sbox6</b>															
12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

<b>Sbox7</b>															
4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	4	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

Sbox8															
13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

**Bảng 3.1.** 8 Sbox của thuật toán DES

### 3.2.1.2. P-box (Permutation box)

Là bước hoán vị bit sau khi dữ liệu đi qua S-box, dữ liệu 32 bit được hoán vị lại theo một bảng cố định. Mục đích nhằm lan truyền sự thay đổi của một bit đầu vào đến nhiều bit đầu ra, góp phần khuếch tán thông tin và tăng tính hỗn loạn.

Bảng P			
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

**Bảng 3.2.** Bảng P

### 3.2.1.3. Bảng mở rộng E

Thành phần này mở rộng khối 32 bit thành 48 bit bằng cách lặp lại một số bit theo bảng mở rộng. Mục tiêu việc mở rộng là tăng độ phức tạp và tạo điều kiện cho XOR với khóa con.

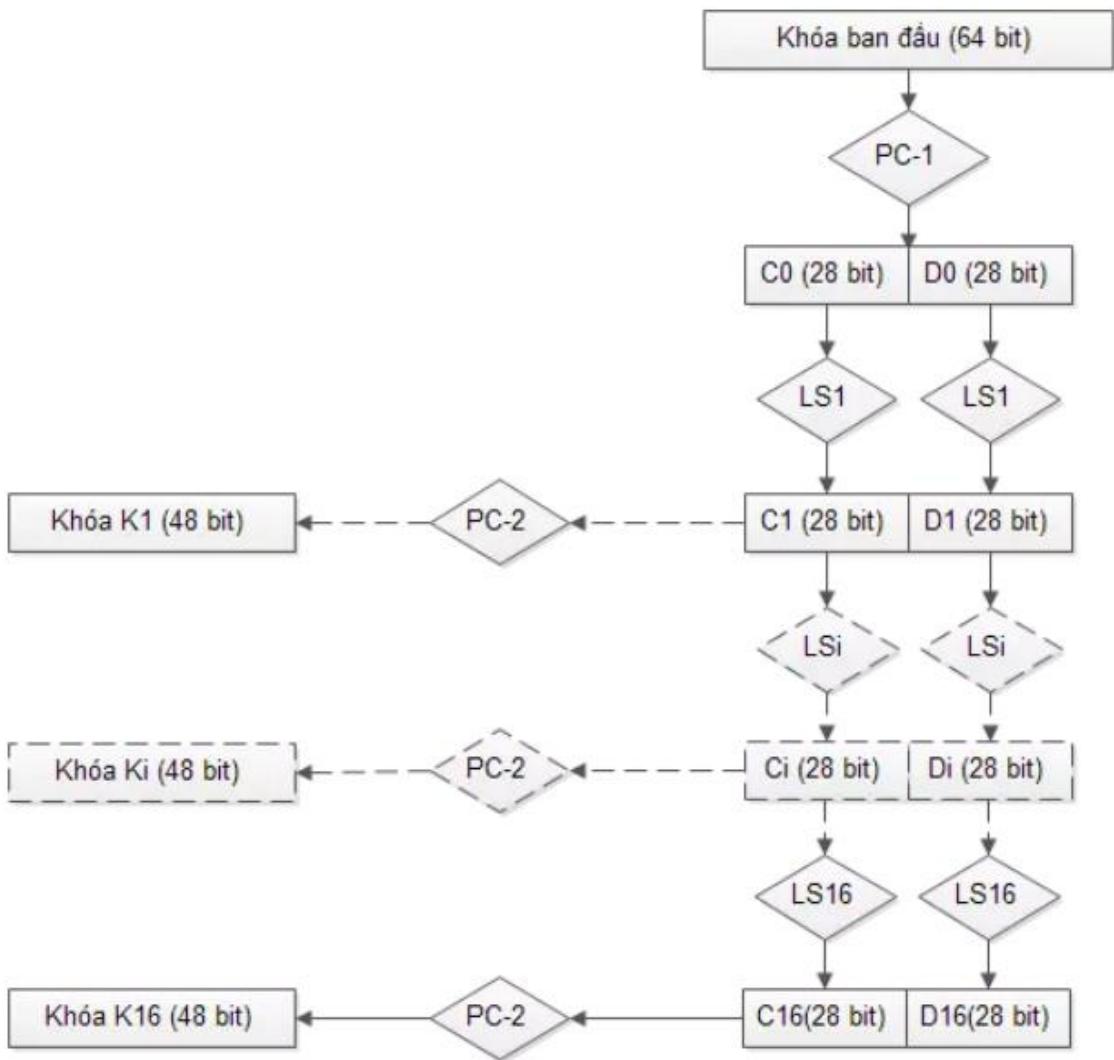
<b>Bảng mở rộng E</b>					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

**Bảng 3.3.** Bảng mở rộng E

### 3.2.2. Khóa con (Subkeys)

Khóa chính ban đầu có độ dài 64 bit, tuy nhiên chỉ có 56 bit thực sự được sử dụng (8 bit còn lại dùng để kiểm tra chẵn lẻ). Từ khóa chính, thuật toán tạo ra 16 khóa con, mỗi khóa có độ dài 48 bit, dùng cho từng vòng lặp. Việc sử dụng khóa con khác nhau ở mỗi vòng giúp tăng cường tính bảo mật và làm cho quá trình mã hóa trở nên khó đoán.

Tóm lại: Từ khóa 64 bit → bỏ 8 bit parity → chia và dịch vòng sinh ra 16 khóa 48 bit.



**Hình 3.3.** Sơ đồ quá trình sinh khóa con (1)

Bảng dịch bit của 16 vòng khóa con như sau:

Vòng	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Số bit dịch	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

**Bảng 3.4.** Bảng dịch bit

### **3.2.3. Hoán vị đầu và hoán vị cuối (IP&FP)**

Hai bảng hoán vị này không trực tiếp tham gia vào quá trình mã hóa dữ liệu mà chủ yếu để sắp xếp lại các bit đầu vào và đầu ra. Tuy nhiên, chúng góp phần tạo tính hỗ trợ cho các thành phần chính và giúp phân phối thông tin bit đồng đều hơn trong toàn bộ thuật toán. Tăng tính khó đoán địa chỉ bit, giúp chuỗi bit trông ngẫu nhiên.

Bảng hoán vị khởi tạo đầu và cuối (IP, IP-1) như sau:

<b>Bảng IP</b>							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

<b>Bảng IP-1</b>							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

**Bảng 3.5.** Bảng hoán vị IP và IP-1

### **3.3. QUY TRÌNH MÃ HÓA VÀ GIẢI MÃ TRONG DES**

#### **3.3.1. Khởi tạo và phân bố khóa**

Ví dụ ta có bản mã M và khóa K như sau:

**M: 0123456789ABCDEF (HEX)**

**K: 133457799BBCDFF1 (HEX)**

Ta cần đổi dữ liệu từ HEX sang dạng nhị phân

##### **3.3.1.1. Chuyển đổi dữ liệu sang dạng nhị phân:**

Vì thuật toán DES mã hóa lần lược các khối dữ liệu kèm khóa 64 bit nên nếu quy đổi sang nhị phân không đủ 64 bit thì ta cần padding ( thêm các bit 0 vào sau cho đủ 64 bit), còn nếu dữ liệu hoặc khóa sau khi đổi nhiều hơn 64 bit nhị phân thì chia thành các khối tương ứng độ dài ( ví dụ nếu 68 bit thì chia làm 2 khối, khối đầu 64 bit đầu tiên, khối hai 4 bit còn lại kèm padding vào khối hai cho đủ 64 bit)

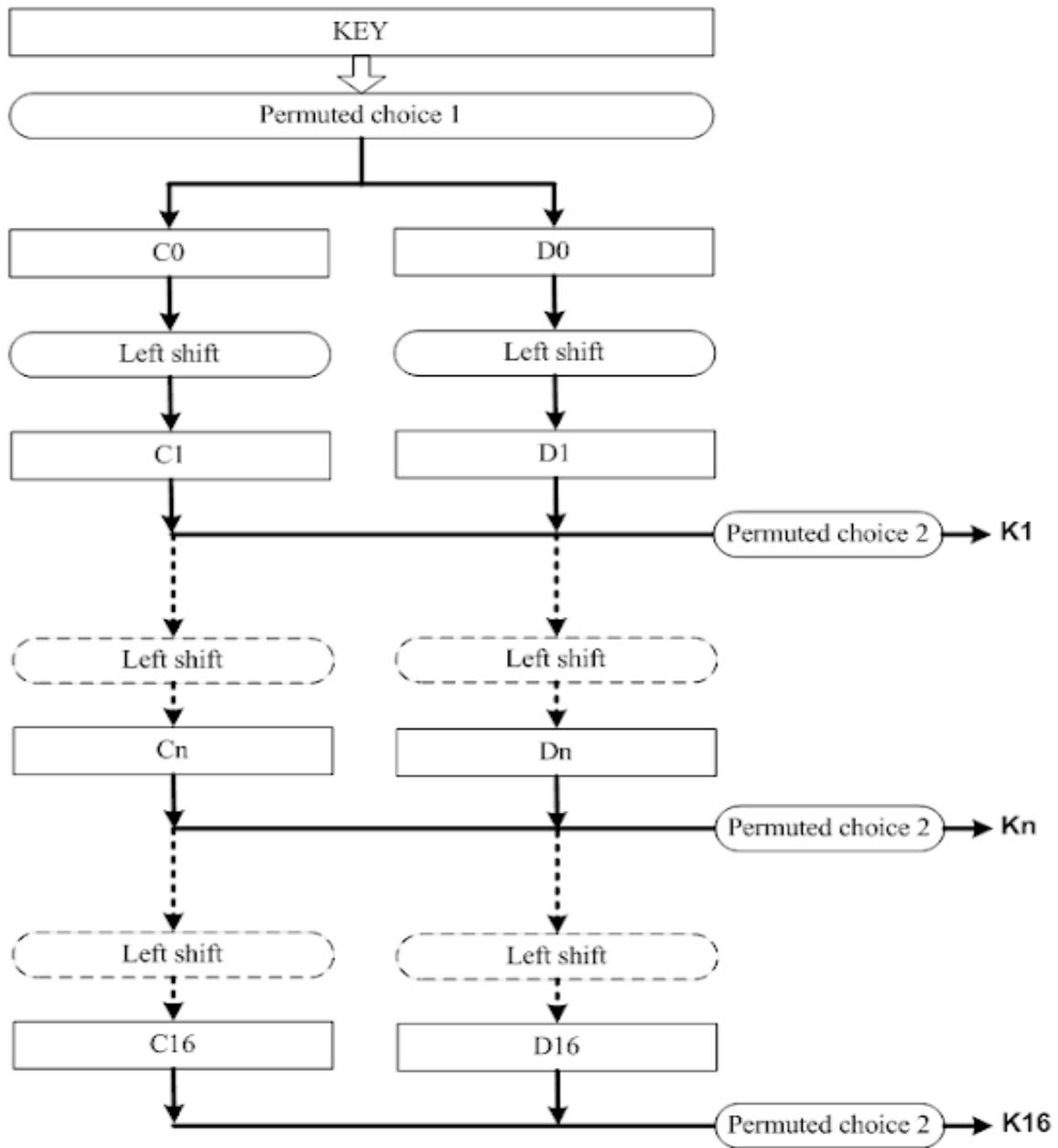
##### **M sau khi chuyển đổi**

0000000100100011010001010110011110001001101010111100110111  
01111

##### **K sau khi chuyển đổi**

0001001100110100010101110111100110011011101111001101111111  
10001

### 3.3.1.2. Tạo khóa con:



Hình 3.4: Sơ đồ tạo khóa con (2)

**Bước 1:** Khóa gốc sẽ được đi qua bảng hoán vị PC-1

Bảng PC-1						
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

**Bảng 3.6.** Bảng hoán vị PC-1

### **Khóa sau khi đi qua bảng hoán vị PC1**

1111000011001100101010101110101010101100110011110001111

Sau khi khóa gốc đi qua bảng PC-1 từ 64 bit sẽ còn lại 56 bit nguyên nhân là:

Chuỗi khóa 64 bit tương đương 8 byte (8 khối mỗi khối 8 bit), mỗi khối 8 bit sẽ chỉ có 7 bit đầu là dữ liệu, bit cuối cùng được gọi là bit kiểm tra chẵn lẻ. Bit này sẽ kiểm tra có lỗi trong khi truyền hay không.

**Ví dụ:** 00010011 có 7 bit đầu là 0001001, khối này có 2 bit 1 -> chẵn nên bit thứ 8 sẽ là 1.

Trong lúc truyền dữ liệu giả sử bit đầu bị sai trở thành 1001001, ta sẽ thấy có 3 bit 1 (thì bit thứ 8 đáng lẽ phải là 0 nhưng ở đây là 1) -> dữ liệu sai

**Bước 2:** Sau khi đi qua bảng PC1, khóa 56 bit sẽ được chia thành 2 khối C0 và D0, mỗi khối 28 bit ta thu được:

**C0:** 1111000011001100101010101111

**D0:** 0101010101100110011110001111

Tiến hành dịch bit C0 và D0 theo bảng dịch bit ta sẽ thu được (C1,D1)...(C16,D16), sau khi dịch gộp C và D lại

**C1:** 111000011001100101010101011111

**D1:** 1010101011001100111100011110

**C1+D1:**

**1110000110011001010101011111010101011001100111100011110**

**C2:** 110000110011001010101010111111

**D2:** 0101010110011001111000111101

**C2+D2:**

**1100001100110010101010111110101010110011001111000111101**

**C3:** 000011001100101010101111111

**D3:** 0101011001100111100011110101

**C3+D3:**

**0000110011001010101011111110101011001100111100011110101**

**C4:** 001100110010101010111111100

**D4:** 0101100110011110001111010101

**C4+D4:**

**0011001100101010101111111000101100110011110001111010101**

**C5:** 110011001010101011111110000

**D5:** 0110011001111000111101010101

**C5+D5:**

**11001100101010111111100000110011001111000111101010101**

**C6:** 0011001010101111111000011

**D6:** 1001100111100011110101010101

**C6+D6:**

**00110010101011111110000111001100111100011110101010101**

**C7:** 1100101010111111100001100

**D7:** 0110011110001111010101010110

**C7+D7:**

**11001010101111111000011000110011110001111010101010110**

**C8:** 001010101011111110000110011

**D8:** 100111100011101010101011001

**C8+D8:**

**0010101010111111000011001110011100011101010101011001**

**C9:** 0101010101111110000110011

**D9:** 0011110001110101010110011

**C9+D9:**

**0101010101111110000110011000111000111010101010110011**

**C10:** 01010101111110000110011001

**D10:** 1111000111010101011001100

**C10+D10:**

**0101010111111000011001100111100011101010101011001100**

**C11:** 01010111111000011001100101

**D11:** 1100011101010101100110011

**C11+D11:**

**01010111111000011001100101110001110101010101100110011**

**C12:** 01011111100001100110010101

**D12:** 0001110101010110011001111

**C12+D12:**

**010111111000011001100101010001110101010110011001111**

**C13:** 011111110000110011001010101

**D13:** 0111010101011001100111100

**C13+D13:**

**011111110000110011001010101011101010101011001100111100**

**C14:** 11111110000110011001010101

**D14:** 11101010101100110011110001

**C14+D14:**

**1111111000011001100101010101110101010101100110011110001**

**C15:** 1111100001100110010101010111

**D15:** 1010101010110011001111000111

**C15+D15:**

**111100001100110010101010111010101011001100111000111**

**C16:** 11110000110011001010101011110101011001100111000111

**D16:** 010101010110011001110001111

**C16+D16:**

**1111000011001100101010101110101010110011001110001111**

**Bước 3:** Sau khi dịch bit và có các chuỗi C+D thì lần lượt đem các chuỗi này qua hoán vị theo bảng PC-2 để loại bỏ và chỉ giữ lại 48 bit để làm các khóa con:

Bảng PC-2					
14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

**Bảng 3.7.** Bảng hoán vị PC-2

**K1:** 00011011000000101101111111000111000001110010

**K2:** 0111001101011101101100111011011100100111100101

**K3:** 0101010111111001000101001000010110011110011001

**K4:** 011100101011011101011011011001101100100011101

**K5:** 01111100111011000000011111010110101001110101000

**K6:** 011000110100101001111001010000011101100101111

**K7:** 11101100100001001011011111101100001100010111100

**K8:** 11110111100010100011101011000001001110111111011

**K9:** 1110000011011011110101111101101110011110000001

**K10:** 101100011111001101000111101110100100011001001111

**K11:** 0010000101011111101001111011101101001110000110

**K12:** 01110101011100011111010110010100011001111101001

**K13:** 1001011110001011101000111110101011101001000001

**K14:** 01011110100001101101111100101110011100111010

**K15:** 1011111100100011000110100111101001111100001010

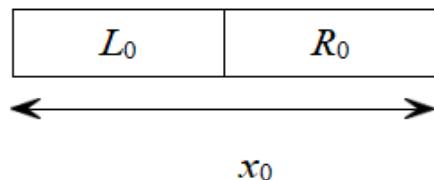
**K16:** 11001011001111011000101100001110000101111110101

Đây là 16 khóa con sẽ sử dụng trong 16 vòng mã hóa của thuật toán DES

### 3.3.2. Quy trình mã hóa trong DES

DES dựa trên cấu trúc Feistel bao gồm nhiều vòng xử lý liên tiếp. Quy trình mã hóa trong DES gồm ba giai đoạn chính sau:

1. Tạo dãy 64 bit  $x_0$  bằng cách hoán vị  $x$  theo hoán vị IP (Initial Permutation). Biểu diễn  $x_0 = IP(x) = L_0R_0$ , trong đó  $L_0$  gồm 32 bit đầu,  $R_0$  gồm 32 bit cuối



**Hình 3.5.** Biểu diễn chuỗi 64 bit  $x_0$  thành 2 phần L và R

2. Thực hiện 16 vòng lặp từ 64 bit thu được và 56 bit của khoá k (chỉ sử dụng 48 bit của khoá k trong mỗi vòng lặp). 64 bit kết quả thu được qua mỗi vòng lặp sẽ là đầu vào cho vòng lặp sau. Các cặp từ 32 bit  $L_i, R_i$  (với  $1 \leq i \leq 16$ ) được xác định theo quy tắc sau:

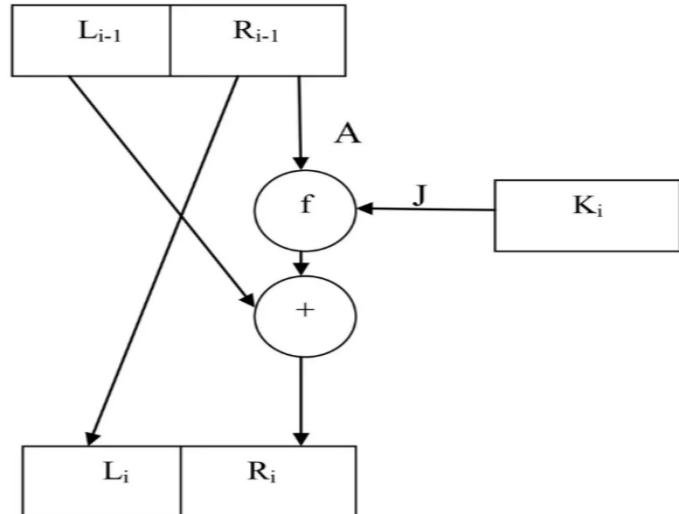
- Mỗi vòng i ( $1 \rightarrow 16$ ):
  - $L[i] = R[i-1]$
  - $R[i] = L[i-1] \text{ XOR } f(R[i-1], K[i])$

với  $\oplus$  biểu diễn phép toán XOR theo modulo 2 của hai dãy bit, K1, K2, ..., K16 là các dãy 48 bit phát sinh từ khóa K cho trước (Trên thực tế, mỗi khóa Ki được phát sinh bằng cách hoán vị các bit trong khóa K cho trước).

i	IP(M)	
	Li	Ri
0	XXXXXXXX	XXXXXXXX
1	$L_1 = R_0$	$R_1 = L_0 \oplus f(R_0, K_1)$
2	$L_2 = R_1$	$R_2 = L_0 \oplus f(R_1, K_2)$
3	$L_3 = R_2$	$R_3 = L_0 \oplus f(R_2, K_3)$
	...	...
16	$L_{16} = R_{15}$	$R_{16} = L_{15} \oplus f(R_{15}, K_{16})$

**Bảng 3.8.** Quy trình mã hóa DES

3. Ghép và hoán vị ngược (Final Permutation -  $IP^{-1}$ ): Sau 16 vòng, hai nửa  $L_{16}$  và  $R_{16}$  hoán vị rồi gộp lại, sau đó áp dụng  $IP^{-1}$  để cho ra ciphertext.



**Hình 3.6.** Một vòng (vòng thứ  $i$ ) của DES

### 3.3.3. Quy trình giải mã trong DES:

DES sử dụng cùng cấu trúc cho mã hóa và giải mã nên các bước của quá trình giải mã dữ liệu được thực hiện tương tự như quá trình mã hóa dữ liệu. Tuy nhiên:

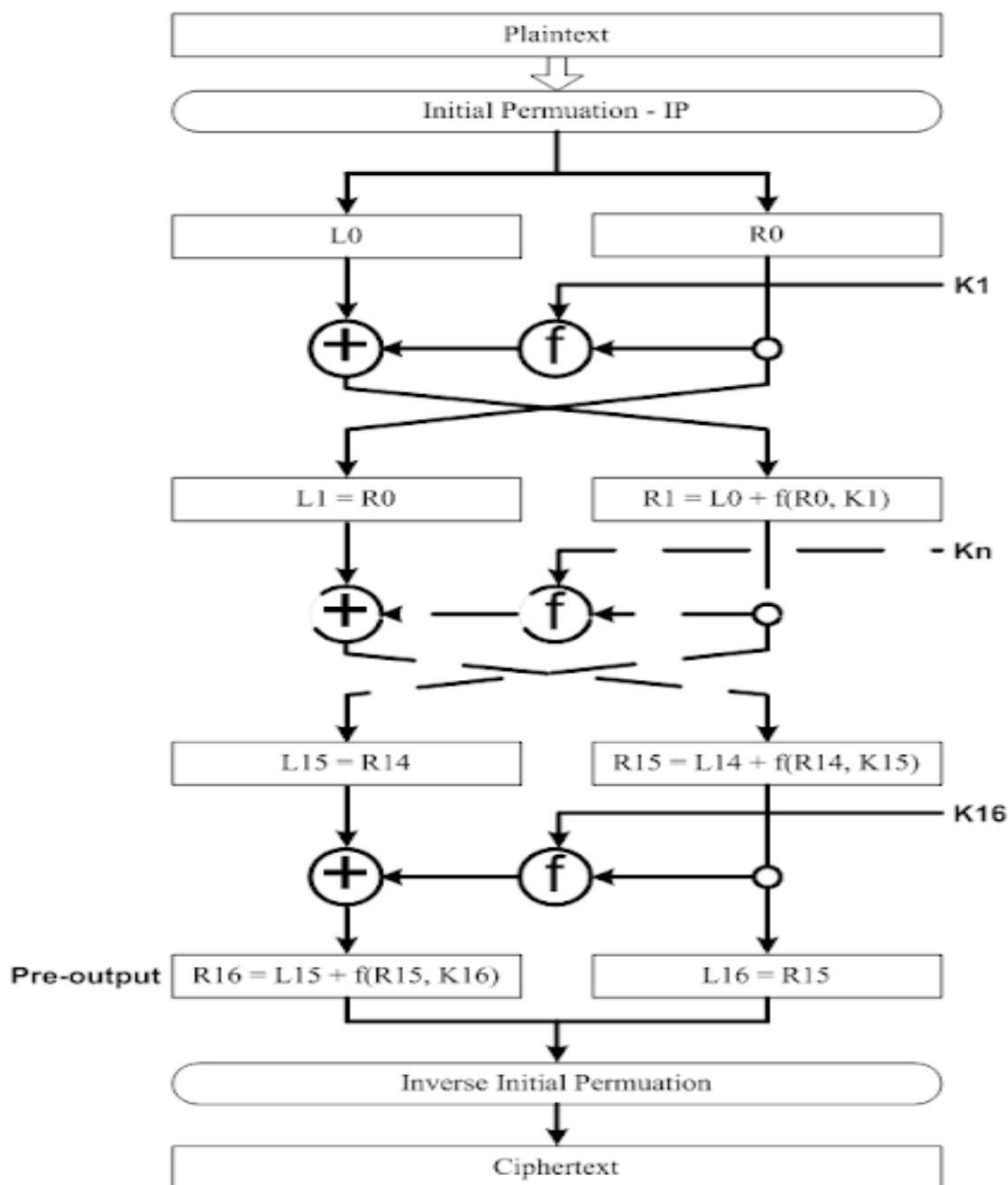
- Trong quá trình giải mã có một số thay đổi là đầu vào lúc này là dữ liệu cần giải mã (ciphertext) và đầu ra là kết quả giải mã được (plaintext).
- Khóa vòng sử dụng trong các vòng lặp giải mã có thứ tự ngược với quá trình mã hóa. Nghĩa là, tại vòng lặp giải mã đầu tiên, khóa vòng được sử dụng là  $K_{16}$ . Tại vòng lặp giải mã thứ 2, khóa vòng được sử dụng là  $K_{15}$ , và tại vòng lặp giải mã cuối cùng thì khóa vòng được sử dụng là  $K_1$ .

### 3.3.4. Ví dụ thực tiễn:

#### Dữ liệu cần mã hóa

0000000100100011010001010110011100010011010101111001101111  
01111

Theo sơ đồ mã hóa DES



Hình 3.7. Sơ đồ mã hóa DES

## Vòng 1

**Bước 1:** Dữ liệu được hoán vị qua bảng hoán vị khởi tạo IP

Dữ liệu trở thành

11001100000000001100110011111111110000101010101110000101  
01010

**Bước 2:** Chia dữ liệu thành L0 và R0, mỗi khối 32 bit

**L0:** 11001100000000001100110011111111110000101010101110000101

**R0:** 1111000010101010111000010101010

**Bước 3:** Tìm L1 và R1

**L1 = R0 =** 1111000010101010111000010101010

**R1 = L0 + f(R0, K1)**

Để tính R1 ta cần tính hàm  $f(R0, K1)$  và XOR nó với L0

- R0 sẽ được đưa qua bảng E (sẽ còn lại 48 bit) và XOR với khóa K1

**E(R0):** 01111010000101010101011110100001010101010101

**K1:** 0001101100000010111011111111000111000001110010

**XOR(E(R0), K1):**

011000010001011110111010100001100110010100100111

- Sau khi có kết quả XOR của E(R0) và K1, chuỗi sẽ được tách thành 8 phần mỗi phần 6 bit và được thay thế bằng cách tra các Sbox (bit đầu và cuối là dòng, 4 bit giữ là cột)

**Sbox1:** 011000 -> dòng 0, cột 12 -> 0101

**Sbox2:** 010001 -> dòng 1, cột 8 -> 1100

**Sbox3:** 011110 -> dòng 0, cột 15 -> 1000

**Sbox4:** 111010 -> dòng 2, cột 13 -> 0010

**Sbox5:** 100001 -> dòng 3, cột 0 -> 1011

**Sbox6:** 100110 -> dòng 2, cột 3 -> 0101

**Sbox7:** 010100 -> dòng 0, cột 10 -> 1001

**Sbox8:** 100111 -> dòng 3, cột 3 -> 0111

**Gộp lại kết quả các Sbox:**

01011100100000101011010110010111

- Sau khi đi qua các Sbox và gộp lại 1 chuỗi bit hoàn chỉnh, ta sẽ đưa chuỗi trên qua bảng P ta thu được hàm  $f(R0, K1)$

**Sau khi qua bảng P:**

**$f(R0, K1)$ :** 001000110100101010100110111011

- Sau khi có hàm  $f(R0, K1)$  ta XOR với L0 sẽ thu được R1

**Kết quả vòng 1:**

**R1:** 11101111010010100110010101000100

**L1:** 11110000101010101111000010101010

## Vòng 2

**$L2 = R1 = 11101111010010100110010101000100$**

**$R2 = L1 + f(R1, K2)$**

**R1 qua bảng E:**

011101011110101001010100001100001010101000001001

**K2:** 011110011010111011011001110110111100100111100101

**XOR(E(R1), K2):**

000011000100010010001101111010110110001111101100

**Sbox1:** 000011 -> dòng 1, cột 1 -> 1111

**Sbox2:** 000100 -> dòng 0, cột 2 -> 1000

**Sbox3:** 010010 -> dòng 0, cột 9 -> 1101

**Sbox4:** 001101 -> dòng 1, cột 6 -> 0000

**Sbox5:** 111010 -> dòng 2, cột 13 -> 0011

**Sbox6:** 110110 -> dòng 2, cột 11 -> 1010

**Sbox7:** 001111 -> dòng 1, cột 7 -> 1010

**Sbox8:** 101100 -> dòng 2, cột 6 -> 1110

**Gộp lại kết quả các Sbox:**

11111000110100000011101010101110

**Sau khi qua bảng P:**

**f(R1,K2):** 00111001010101100001110100011

Sau khi có hàm f(R1,K2) ta XOR với L1 sẽ thu được R2

**Kết quả vòng 2:**

**R2:** 11001100000000010111011100001001

**L2:** 11101111010010100110010101000100

### **Vòng 3**

**L3 = R2 =** 11001100000000010111011100001001

**R3 = L2 + f(R2,K3)**

**R2 qua bảng E:**

11100101100000000000010101110101110100001010011

**K3:**

010101011111001000101001000010110011110011001

**XOR(E(R2), K3):**

1011000001111100100010001111000001001111001010

**Sbox1:** 101100 -> dòng 2, cột 6 -> 0010

**Sbox2:** 000111 -> dòng 1, cột 3 -> 0111

**Sbox3:** 110010 -> dòng 2, cột 9 -> 0001

**Sbox4:** 001000 -> dòng 0, cột 4 -> 0000

**Sbox5:** 111110 -> dòng 2, cột 15 -> 1110

**Sbox6:** 000010 -> dòng 0, cột 1 -> 0001

**Sbox7:** 011111 -> dòng 1, cột 15 -> 0110

**Sbox8:** 001010 -> dòng 0, cột 5 -> 1111

**Gộp lại kết quả các Sbox:**

00100111000100001110000101101111

**Sau khi qua bảng P:**

**f(R2,K3):** 01001101000101100110111010110000

Sau khi có hàm f(R2,K3) ta XOR với L2 sẽ thu được R3

**Kết quả vòng 3:**

**R3:** 1010001001011100000010111110100

**L3:** 11001100000000010111011100001001

### Vòng 4

**L4 = R3 =** 1010001001011100000010111110100

**R4 = L3 + f(R3,K4)**

**R3 qua bảng E:**

010100000100001011110000000010101111110101001

**K4:**

01110010101011011010110110110011010100011101

**XOR(E(R3), K4):**

0010001011101111001011101101111001001010110100

**Sbox1:** 001000 -> dòng 0, cột 4 -> 0010

**Sbox2:** 101110 -> dòng 2, cột 7 -> 0001

**Sbox3:** 111100 -> dòng 2, cột 14 -> 1110

**Sbox4:** 101110 -> dòng 2, cột 7 -> 1101

**Sbox5:** 110111 -> dòng 3, cột 11 -> 1001

**Sbox6:** 100100 -> dòng 2, cột 2 -> 1111

**Sbox7:** 101010 -> dòng 2, cột 5 -> 0011

**Sbox8:** 110100 -> dòng 2, cột 10 -> 1010

**Gộp lại kết quả các Sbox:**

00100001111011011001111100111010

**Sau khi qua bảng P:**

**f(R3,K4):** 10111011001000110111011101001100

Sau khi có hàm f(R3,K4) ta XOR với L3 sẽ thu được R4

**Kết quả vòng 4:**

**R4:** 011101110010001000000000001000101

**L4:** 1010001001011100000010111110100

## Vòng 5

**L5 = R4** = 01110111001000100000000001000101

**R5 = L4 + f(R4,K5)**

**R4 qua bảng E:**

10111010111010010000010000000000000001000001010

**K5:**

01111100111011000000011111010110101001110101000

**XOR(E(R4), K5):**

11000110000001010000001111010110101000110100010

**Sbox1:** 110001 -> dòng 3, cột 8 -> 0101

**Sbox2:** 100000 -> dòng 2, cột 0 -> 0000

**Sbox3:** 010100 -> dòng 0, cột 10 -> 1100

**Sbox4:** 000011 -> dòng 1, cột 1 -> 1000

**Sbox5:** 111010 -> dòng 2, cột 13 -> 0011

**Sbox6:** 110101 -> dòng 3, cột 10 -> 0001

**Sbox7:** 000110 -> dòng 0, cột 3 -> 1110

**Sbox8:** 100010 -> dòng 2, cột 1 -> 1011

**Gộp lại kết quả các Sbox:**

01010000110010000011000111101011

**Sau khi qua bảng P:**

**f(R4,K5):**

0010100000010011101011011000011

Sau khi có hàm f(R4,K5) ta XOR với L4 sẽ thu được R5

**Kết quả vòng 5:**

**R5:** 1000101001001111010011000110111

**L5:** 01110111001000100000000001000101

## Vòng 6

**L6 = R5 =** 1000101001001111010011000110111

**R6 = L5 + f(R5,K6)**

**R5 qua bảng E:**

1100010101000010010111110100001100000110101111

**K6:**

01100011101001010011110010100000111101100101111

**XOR(E(R5), K6):**

10100110110011101100001100000001011101010000000

**Sbox1:** 101001 -> dòng 3, cột 4 -> 0100

**Sbox2:** 101110 -> dòng 2, cột 7 -> 0001

**Sbox3:** 011101 -> dòng 1, cột 14 -> 1111

**Sbox4:** 100001 -> dòng 3, cột 0 -> 0011

**Sbox5:** 100000 -> dòng 2, cột 0 -> 0100

**Sbox6:** 001011 -> dòng 1, cột 5 -> 1100

**Sbox7:** 101010 -> dòng 2, cột 5 -> 0011

**Sbox8:** 000000 -> dòng 0, cột 0 -> 1101

**Gộp lại kết quả các Sbox:**

0100000111100110100110000111101

**Sau khi qua bảng P:**

**f(R5,K6):**

10011110010001011100110100101100

Sau khi có hàm f(R5,K6) ta XOR với L5 sẽ thu được R6

**Kết quả vòng 6:**

**R6:** 1110100101100111100110101101001

**L6:** 1000101001001111010011000110111

## Vòng 7

**L7 = R6 =** 1110100101100111100110101101001

**R7 = L6 + f(R6,K7)**

**R6 qua bảng E:**

111101010010101100001111110010110101011010011

**K7:**

11101100100001001011011111101100001100010111100

**XOR(E(R6), K7):**

000110011010111101110000010011101100111101111

**Sbox1:** 000110 -> dòng 0, cột 3 -> 0001

**Sbox2:** 011010 -> dòng 0, cột 13 -> 0000

**Sbox3:** 111110 -> dòng 2, cột 15 -> 0111

**Sbox4:** 111000 -> dòng 2, cột 12 -> 0101

**Sbox5:** 000100 -> dòng 0, cột 2 -> 0100

**Sbox6:** 111011 -> dòng 3, cột 13 -> 0000

**Sbox7:** 001111 -> dòng 1, cột 7 -> 1010

**Sbox8:** 101111 -> dòng 3, cột 7 -> 1101

**Gộp lại kết quả các Sbox:**

00010000011101010100000010101101

**Sau khi qua bảng P:**

**f(R6,K7):**

10001100000001010001110000100111

Sau khi có hàm f(R6,K7) ta XOR với L6 sẽ thu được R7

**Kết quả vòng 7:**

**R7:** 00000110010010101011101000010000

**L7:** 1110100101100111100110101101001

## Vòng 8

**L8 = R7 = 00000110010010101011101000010000**

**R8 = L7 + f(R7,K8)**

**R7 qua bảng E:**

000000001100001001010101011110100000010100000

**K8:**

1111011100010100011101011000001001110111111011

**XOR(E(R7), K8):**

111101110100100001101111001111001111011011011

**Sbox1:** 111101 -> dòng 3, cột 14 -> 0110

**Sbox2:** 110100 -> dòng 2, cột 10 -> 1100

**Sbox3:** 100001 -> dòng 3, cột 0 -> 0001

**Sbox4:** 101111 -> dòng 3, cột 7 -> 1000

**Sbox5:** 100111 -> dòng 3, cột 3 -> 0111

**Sbox6:** 100111 -> dòng 3, cột 3 -> 1100

**Sbox7:** 101101 -> dòng 3, cột 6 -> 1010

**Sbox8:** 011011 -> dòng 1, cột 13 -> 1110

**Gộp lại kết quả các Sbox:**

0110110000011000011110010101110

**Sau khi qua bảng P:**

**f(R7,K8):**

0011110000001110100001101111001

Sau khi có hàm f(R7,K8) ta XOR với L7 sẽ thu được R8

**Kết quả vòng 8:**

**R8:** 11010101011010010100101110010000

**L8:** 00000110010010101011101000010000

## Vòng 9

**L9 = R8 =** 11010101011010010100101110010000

**R9 = L8 + f(R8,K9)**

**R8 qua bảng E:**

0110101010101101010010100101011110010100001

**K9:**

1110000011011011110101111011011110011110000001

**XOR(E(R8), K9):**

100010100111000010111001010010001001101100100000

**Sbox1:** 100010 -> dòng 2, cột 1 -> 0001

**Sbox2:** 100111 -> dòng 3, cột 3 -> 0001

**Sbox3:** 000010 -> dòng 0, cột 1 -> 0000

**Sbox4:** 111001 -> dòng 3, cột 12 -> 1100

**Sbox5:** 010010 -> dòng 0, cột 9 -> 0101

**Sbox6:** 001001 -> dòng 1, cột 4 -> 0111

**Sbox7:** 101100 -> dòng 2, cột 6 -> 0111

**Sbox8:** 100000 -> dòng 2, cột 0 -> 0111

**Gộp lại kết quả các Sbox:**

00010001000011000101011101110111

**Sau khi qua bảng P:**

**f(R8,K9):**

0010001000110110011110001101010

Sau khi có hàm f(R8,K9) ta XOR với L8 sẽ thu được R9

**Kết quả vòng 9:**

**R9:** 0010010001111001100011001111010

**L9:** 11010101011010010100101110010000

## Vòng 10

**L10 = R9 = 0010010001111001100011001111010**

**R10 = L9 + f(R9,K10)**

**R9 qua bảng E:**

00010000100000111111001011000001100001111110100

**K10:**

1011000111100110100011101110100100011001001111

**XOR(E(R9), K10):**

10100001011100001011110110110101000010110111011

**Sbox1:** 101000 -> dòng 2, cột 4 -> 1101

**Sbox2:** 010111 -> dòng 1, cột 11 -> 1010

**Sbox3:** 000010 -> dòng 0, cột 1 -> 0000

**Sbox4:** 111110 -> dòng 2, cột 15 -> 0100

**Sbox5:** 110110 -> dòng 2, cột 11 -> 0101

**Sbox6:** 101000 -> dòng 2, cột 4 -> 0010

**Sbox7:** 010110 -> dòng 0, cột 11 -> 0111

**Sbox8:** 111011 -> dòng 3, cột 13 -> 0101

**Gộp lại kết quả các Sbox:**

11011010000001000101001001110101

**Sau khi qua bảng P:**

**f(R9,K10):**

0110001010111001001110000100010

Sau khi có hàm f(R9,K10) ta XOR với L9 sẽ thu được R10

**Kết quả vòng 10:**

**R10:** 101101111010101101011110110010

**L10:** 0010010001111001100011001111010

## Vòng 11

**L11 = R10 = 1011011110101011101011110110010**

**R11 = L10 + f(R10,K11)**

**R10 qua bảng E:**

010110101111110101010111101010111110110100101

**K11:**

0010000101011111101001111011101101001110000110

**XOR(E(R10), K11):**

01111011101000010111000001101000010111000100011

**Sbox1:** 011110 -> dòng 0, cột 15 -> 0111

**Sbox2:** 111010 -> dòng 2, cột 13 -> 0011

**Sbox3:** 000101 -> dòng 1, cột 2 -> 0000

**Sbox4:** 111000 -> dòng 2, cột 12 -> 0101

**Sbox5:** 001101 -> dòng 1, cột 6 -> 1101

**Sbox6:** 000010 -> dòng 0, cột 1 -> 0001

**Sbox7:** 111000 -> dòng 2, cột 12 -> 0000

**Sbox8:** 100011 -> dòng 3, cột 1 -> 0001

**Gộp lại kết quả các Sbox:**

01110011000001011101000100000001

**Sau khi qua bảng P:**

**f(R10,K11):**

11100001000001001111101000000010

Sau khi có hàm f(R10,K11) ta XOR với L10 sẽ thu được R11

**Kết quả vòng 11:**

**R11:** 110001010111000001110001111000

**L11:** 1011011110101011101011110110010

## Vòng 12

**L12 = R11 = 11000101011110000011110001111000**

**R12 = L11 + f(R11, K12)**

**R11 qua bảng E:**

011000001010101111100000001111100000111110001

**K12:**

01110101011100011111010110010100011001111101001

**XOR(E(R11), K12):**

00010101110110100000010110001011110010000011000

**Sbox1:** 000101 -> dòng 1, cột 2 -> 0111

**Sbox2:** 011101 -> dòng 1, cột 14 -> 1011

**Sbox3:** 101000 -> dòng 2, cột 4 -> 1000

**Sbox4:** 000101 -> dòng 1, cột 2 -> 1011

**Sbox5:** 100010 -> dòng 2, cột 1 -> 0010

**Sbox6:** 111110 -> dòng 2, cột 15 -> 0110

**Sbox7:** 010000 -> dòng 0, cột 8 -> 0011

**Sbox8:** 011000 -> dòng 0, cột 12 -> 0101

**Gộp lại kết quả các Sbox:**

01111011100010110010011000110101

**Sau khi qua bảng P:**

**f(R11, K12):**

1100001001101000110011111101010

Sau khi có hàm f(R11, K12) ta XOR với L11 sẽ thu được R12

**Kết quả vòng 12:**

**R12:** 01110101101111010001100001011000

**L12:** 110001010111000001110001111000

### Vòng 13

**L13 = R12 = 01110101101111010001100001011000**

**R13 = L12 + f(R12, K13)**

**R12 qua bảng E:**

001110101011110111111010100011110000001011110000

**K13:**

10010111100010111010001111101011101001000001

**XOR(E(R12), K13):**

10101101011110000010101101110101101100010110001

**Sbox1:** 101011 -> dòng 3, cột 5 -> 1001

**Sbox2:** 010111 -> dòng 1, cột 11 -> 1010

**Sbox3:** 100000 -> dòng 2, cột 0 -> 1101

**Sbox4:** 101011 -> dòng 3, cột 5 -> 0001

**Sbox5:** 011101 -> dòng 1, cột 14 -> 1000

**Sbox6:** 011011 -> dòng 1, cột 13 -> 1011

**Sbox7:** 100010 -> dòng 2, cột 1 -> 0100

**Sbox8:** 110001 -> dòng 3, cột 8 -> 1111

**Gộp lại kết quả các Sbox:**

10011010110100011000101101001111

**Sau khi qua bảng P:**

**f(R12, K13):**

11011101101110110010100100100010

Sau khi có hàm f(R12, K13) ta XOR với L12 sẽ thu được R13

**Kết quả vòng 13:**

**R13:** 00011000110000110001010101011010

**L13:** 01110101101111010001100001011000

## Vòng 14

**L14 = R13 = 00011000110000110001010101011010**

**R14 = L13 + f(R13,K14)**

**R13 qua bảng E:**

0000111100010110000001101000101010101011110100

**K14:**

01011111010000111011011111100101110011100111010

**XOR(E(R13), K14):**

010100000101010110110001011110000100110111001110

**Sbox1:** 010100 -> dòng 0, cột 10 -> 0110

**Sbox2:** 000101 -> dòng 1, cột 2 -> 0100

**Sbox3:** 010110 -> dòng 0, cột 11 -> 0111

**Sbox4:** 110001 -> dòng 3, cột 8 -> 1001

**Sbox5:** 011110 -> dòng 0, cột 15 -> 1001

**Sbox6:** 000100 -> dòng 0, cột 2 -> 1010

**Sbox7:** 110111 -> dòng 3, cột 11 -> 1111

**Sbox8:** 001110 -> dòng 0, cột 7 -> 0001

**Gộp lại kết quả các Sbox:**

01100100011110011001101011110001

**Sau khi qua bảng P:**

**f(R13,K14):**

10110111001100011000111001010101

Sau khi có hàm f(R13,K14) ta XOR với L13 sẽ thu được R14

**Kết quả vòng 14:**

**R14:** 11000010100011001001011000001101

**L14:** 00011000110000110001010101011010

## Vòng 15

**L15 = R14 = 11000010100011001001011000001101**

**R15 = L14 + f(R14,K15)**

**R14 qua bảng E:**

1110000001010100010110010100101100000001011011

**K15:**

1011111100100011000110100111101001111100001010

**XOR(E(R14), K15):**

0101111100010111010100011101111111101010001

**Sbox1:** 010111 -> dòng 1, cột 11 -> 1011

**Sbox2:** 111100 -> dòng 2, cột 14 -> 0010

**Sbox3:** 010111 -> dòng 1, cột 11 -> 1110

**Sbox4:** 010100 -> dòng 0, cột 10 -> 1000

**Sbox5:** 011101 -> dòng 1, cột 14 -> 1000

**Sbox6:** 111111 -> dòng 3, cột 15 -> 1101

**Sbox7:** 111101 -> dòng 3, cột 14 -> 0011

**Sbox8:** 010001 -> dòng 1, cột 8 -> 1100

**Gộp lại kết quả các Sbox:**

10110010111010001000110100111100

**Sau khi qua bảng P:**

**f(R14,K15):**

01011011100000010010011101101110

Sau khi có hàm f(R14,K15) ta XOR với L14 sẽ thu được R15

**Kết quả vòng 15:**

**R15:** 01000011010000100011001000110100

**L15:** 11000010100011001001011000001101

## Vòng 16

**L16 = R15 = 01000011010000100011001000110100**

**R16 = L15 + f(R15,K16)**

**R15 qua bảng E:**

001000000110101000000100000110100100000110101000

**K16:**

11001011001111011000101100001110000101111110101

**XOR(E(R15), K16):**

11101011010101110001111000101000101011001011101

**Sbox1:** 111010 -> dòng 2, cột 13 -> 1010

**Sbox2:** 110101 -> dòng 3, cột 10 -> 0111

**Sbox3:** 011110 -> dòng 0, cột 15 -> 1000

**Sbox4:** 001111 -> dòng 1, cột 7 -> 0011

**Sbox5:** 000101 -> dòng 1, cột 2 -> 0010

**Sbox6:** 000101 -> dòng 1, cột 2 -> 0100

**Sbox7:** 011001 -> dòng 1, cột 12 -> 0010

**Sbox8:** 011101 -> dòng 1, cột 14 -> 1001

**Gộp lại kết quả các Sbox:**

10100111100000110010010000101001

**Sau khi qua bảng P:**

**f(R15,K16):** 11001000110000000100111110011000

Sau khi có hàm f(R15,K16) ta XOR với L15 sẽ thu được R16

**Kết quả vòng 16:**

**R16:** 00001010010011001101100110010101

**L16:** 01000011010000100011001000110100

**Gộp R16 + L16 lại**

00001010010011001101100110010101010000110100001000110010001  
10100

**Sau khi gộp lại chuỗi được đưa qua bảng IP-1 để ra kết quả cuối**

**Kết quả sau khi thực hiện 16 vòng mã hóa là:**

1000010111101000000100110101010000001110000101010110100000  
00101

### **3.4. ƯU NHƯỢC ĐIỂM VÀ CÁC YẾU TỐ KHÁC CỦA DES**

#### **3.4.1. Ưu điểm của DES**

Cấu trúc đơn giản và dễ hiểu: DES sử dụng cấu trúc Feistel với 16 vòng lặp, có tính lặp lại và logic cao. Nhờ đó, quá trình triển khai, giảng dạy và minh họa trở nên dễ tiếp cận hơn.

Chuẩn hóa rộng rãi và tốc độ nhanh: Là một trong những thuật toán đầu tiên được chuẩn hóa (bởi NIST năm 1977), DES đã được sử dụng rộng rãi trong chính phủ, doanh nghiệp hoặc công nghiệp trong nhiều thập kỷ và vì là mã hóa đối xứng nên nó cũng có tốc độ xử lý rất nhanh.

Hiệu suất cao trong phần cứng: DES được thiết kế để thực thi hiệu quả trên phần cứng, đặc biệt là các thiết bị có tài nguyên hạn chế, giúp tăng tốc độ xử lý trong môi trường công nghiệp.

Cơ sở cho các thuật toán nâng cao hơn: DES đặt nền móng cho nhiều thuật toán sau này như Triple DES (3DES), giúp mở rộng khả năng bảo mật trong khi vẫn giữ cấu trúc cốt lõi.

#### **3.4.2. Nhược điểm của DES**

Không còn được chấp nhận là chuẩn hiện đại: Mặc dù DES từng là một chuẩn mã hóa mạnh mẽ, nhưng hiện nay nó không còn được công nhận là an toàn và đã bị loại khỏi nhiều tiêu chuẩn bảo mật, hạn chế khả năng ứng dụng trong các hệ thống hiện đại.

**Độ dài khóa ngắn và thiết kế cũ:** Với khóa chỉ dài 56 bit và kiến trúc thiết kế từ những năm 1970, DES không còn đáp ứng được các yêu cầu bảo mật trước các tấn công hiện đại. Việc này làm giảm tính hiệu quả và độ tin cậy khi triển khai trong các môi trường yêu cầu bảo mật cao.

Từ năm 2001, thuật toán này chính thức bị thay thế bởi AES (Advanced Encryption Standard) – một thuật toán mạnh hơn với khả năng hỗ trợ khóa dài hơn và hiệu suất cao hơn. Điều này làm giảm tính ứng dụng thực tiễn của DES trong các hệ thống bảo mật ngày nay.

### **3.5. TÍCH HỢP THUẬT TOÁN DIFFIE-HELLMAN, RSA VÀ MD5 ĐỂ TĂNG CƯỜNG BẢO MẬT CHO HỆ THỐNG MÃ HÓA DES**

#### **3.5.1. Lý do sử dụng các thuật toán hỗ trợ trong hệ thống DES:**

Thuật toán DES là một thuật toán mã hóa đối xứng, trong đó cả hai bên gửi và nhận đều sử dụng cùng một khóa để mã hóa và giải mã dữ liệu. Tuy nhiên, điều này dẫn đến một nhược điểm lớn là nếu khóa bị lộ trong quá trình truyền, toàn bộ thông tin sẽ không còn an toàn. Một vấn đề đặt ra là làm thế nào để chia sẻ khóa an toàn giữa hai bên mà không bị lộ thông qua kênh truyền thông công cộng?

Vì vậy, để khắc phục vấn đề đó việc kết hợp DES với các kỹ thuật mật mã hiện đại như trao đổi khóa Diffie-Hellman, ký số bằng RSA và băm chuẩn hóa bằng MD5 là một hướng đi hợp lý nhằm nâng cao tính bảo mật tổng thể của hệ thống.

Cụ thể là:

Diffie-Hellman cho phép các bên trao đổi khóa bí mật qua kênh không an toàn mà không cần truyền khóa trực tiếp.

RSA giải quyết điểm yếu của DH bằng cách ký số các giá trị trao đổi, chống tấn công giả mạo.

MD5 dùng để chuẩn hóa khóa đầu ra của DH thành một chuỗi có độ dài phù hợp để sử dụng với DES.

Sự kết hợp này giúp tận dụng ưu điểm của từng thuật toán và giảm thiểu các rủi ro bảo mật trong quá trình mã hóa và truyền dữ liệu.

### **3.5.2. Mô hình trao đổi khóa của Diffie-Hellman:**

Năm 1976, Whitfield Diffie và Martin Hellman đã đưa ra một giao thức để trao đổi các giá trị khóa quy ước giữa các đối tác trên đường truyền có độ bảo mật trung bình. Sự ra đời của giao thức trao đổi khóa Diffie-Hellman được xem là bước mở đầu cho lĩnh vực mã hóa khóa công cộng.

Giao thức này dựa trên nguyên lý của bài toán logarit rời rạc trên trường số nguyên hữu hạn. Các thao tác thực hiện trao đổi khóa Diffie-Hellman giữa hai đối tác A và B như sau:

1. A và B thống nhất các giá trị  $g$  và số nguyên tố  $p < g$
2. A chọn một số ngẫu nhiên  $m$ . A tính giá trị  $Q_A = g^m \text{ mod } p$  và gửi  $Q_A$  cho B
3. B chọn một số ngẫu nhiên  $n$ . B tính giá trị  $Q_B = g^n \text{ mod } p$  và gửi  $Q_B$  cho A
4. A nhận được  $Q_B$  và tính giá trị  $k = (Q_B)^m \text{ mod } p$
5. B nhận được  $Q_A$  và tính giá trị  $k = (Q_A)^n \text{ mod } p$

-> A và B có  $k$  chính là giá trị bí mật được quy ước chung gọi là khóa chung.

A				B		
Bí mật	Công khai	Tính	Gửi	Tính	Công khai	Bí mật
m	p, g		p, g ->			n
m	p, g, Q <sub>A</sub>	$Q_A = g^m \pmod{p}$	Q <sub>A</sub> ->		p, g	n
m	p, g, Q <sub>A</sub>		<- Q <sub>B</sub>	$Q_B = g^n \pmod{p}$	p, g, Q <sub>A</sub> , Q <sub>B</sub>	n
m, k	p, g, Q <sub>A</sub> , Q <sub>B</sub>	$k = (Q_B)^m \pmod{p}$		$k = (Q_A)^n \pmod{p}$	p, g, Q <sub>A</sub> , Q <sub>B</sub>	n, k

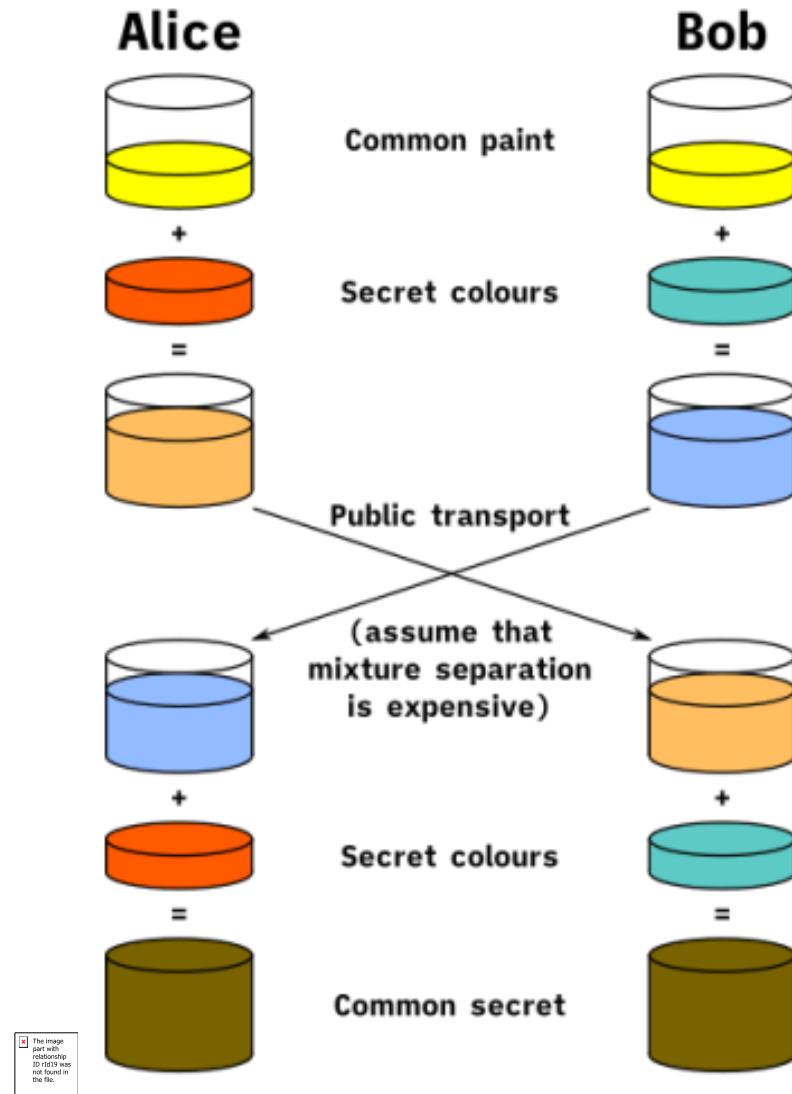
**Bảng 3.9.** Bảng quy trình trao đổi khóa Diffie-Hellman

#### Ví dụ về mô hình trao đổi khóa Diffie-Hellman:

Ý tưởng trao đổi khóa của Diffie-Hellman có thể mô tả thông qua việc trao đổi màu sơn giữa Alice và Bob:

- ❖ Đầu tiên Alice và Bob trộn màu đã biết chung (màu vàng) với màu bí mật riêng của mỗi người. Sau đó, mỗi người chuyển hỗn hợp của mình tới người kia thông qua một kênh vận chuyển công cộng.
- ❖ Khi nhận được hỗn hợp của người kia, mỗi người sẽ trộn thêm với màu bí mật của riêng mình và nhận được hỗn hợp cuối cùng.
- ❖ Hỗn hợp sơn cuối cùng là hoàn toàn giống nhau cho cả hai người và chỉ có riêng hai người biết
- ❖ Mẫu chốt ở đây là đối với một người ngoài sẽ rất khó (về mặt tính toán) cho họ để tìm ra được hỗn hợp bí mật chung của hai người

(nghĩa là hỗn hợp cuối cùng). Alice và Bob sẽ sử dụng hỗn hợp bí mật chung này để mã hóa và giải mã dữ liệu truyền trên kênh công cộng



**Hình 3.8.** Mô hình ví dụ trao đổi màu sơn giữa Alice và Bob

### 3.5.3. Tăng cường xác minh danh tính bằng thuật toán RSA

RSA là một trong những thuật toán mã hóa bất đối xứng phổ biến nhất, cho phép không chỉ mã hóa mà còn ký số và xác minh chữ ký, tức là xác thực danh tính người gửi. Đây là giải pháp hiệu quả để chống lại tấn công Man-in-the-Middle (một nhược điểm tiềm ẩn trong Diffie-Hellman).

## Ứng dụng trong hệ thống:

- Mỗi bên tham gia (A và B) đều có một cặp khóa RSA (khóa công khai và khóa bí mật).
- Khi A gửi giá trị  $Q_A = g^m \text{ mod } p$  (trong DH) cho B, A sẽ ký số giá trị này bằng khóa bí mật RSA của mình.
- B khi nhận được  $Q_A$ , sẽ sử dụng khóa công khai RSA của A để kiểm tra xác minh chữ ký số.
- Nếu xác minh thành công chữ ký hợp lệ, B biết rằng giá trị  $Q_A$  đến từ A, và không bị thay đổi hay giả mạo

⇒ Việc này loại bỏ khả năng giả danh, đảm bảo quá trình trao đổi khóa diễn ra giữa đúng hai bên dự kiến.

### 3.5.4. Chuẩn hóa khóa bằng hàm băm MD5

MD5 (Message Digest Algorithm 5) là một hàm băm mật mã phổ biến, được phát triển bởi Ronald Rivest vào năm 1991. MD5 chuyển đổi một chuỗi đầu vào thành một chuỗi băm có độ dài cố định 128 bit.

Trong hệ thống kết hợp mã hóa DES, Sau khi hai bên tạo ra khóa chung k thông qua Diffie-Hellman, giá trị này có thể là một số nguyên lớn, không theo định dạng tiêu chuẩn. Trong khi đó, DES yêu cầu khóa đầu vào có độ dài chính xác là 56 bit. Do đó, cần phải sử dụng một hàm băm mật mã để chuyển đổi giá trị khóa sang dạng phù hợp.

Do đó, cần phải sử dụng hàm băm MD5 để:

- Băm giá trị khóa chung k thành một chuỗi băm 128 bit.
- Chuyển chuỗi băm thành dạng nhị phân.
- Trích xuất 56 bit từ kết quả nhị phân để dùng làm khóa cho thuật toán DES.

⇒ Nhờ vậy, khóa từ DH được chuẩn hóa, tương thích với yêu cầu kỹ thuật của DES và vẫn đảm bảo tính ngẫu nhiên, khó đoán.

### **3.5.5. Mô hình tích hợp ba lớp bảo mật**

#### **3.5.5.1. Mô tả mô hình tích hợp ba lớp:**

Sự kết hợp giữa các thuật toán tạo nên một mô hình bảo mật gồm nhiều lớp, mỗi lớp đảm nhận một chức năng riêng biệt nhưng hỗ trợ cho nhau, tạo thành hệ thống mã hóa an toàn và hiệu quả hơn.

- ✧ Diffie-Hellman: Thiết lập khóa bí mật chung mà không cần truyền khóa trực tiếp.
- ✧ RSA: Xác minh danh tính người gửi/nhận trong quá trình trao đổi khóa, ngăn chặn MITM.
- ✧ MD5: Chuẩn hóa khóa thành định dạng phù hợp cho thuật toán DES.
- ✧ DES: Mã hóa và giải mã nội dung thực tế với hiệu suất cao.

Việc tích hợp ba lớp kỹ thuật như trên mang lại mức độ bảo mật cao hơn nhiều so với việc chỉ sử dụng thuật toán DES đơn lẻ. Nó vừa đảm bảo trao đổi khóa an toàn, xác minh được danh tính hai bên, vừa tạo ra khóa mã hóa phù hợp với DES.

#### **3.5.5.2. Lợi ích của mô hình tích hợp DES-DH-RSA-MD5:**

Tăng cường độ an toàn cho quá trình trao đổi khóa: Thay vì sử dụng khóa cố định hoặc trao đổi thủ công, việc áp dụng Diffie-Hellman giúp hai bên tạo ra một khóa chung một cách tự động và bảo mật, đảm bảo rằng không ai khác có thể dễ dàng đoán được khóa.

Ngăn chặn tấn công giả mạo và trung gian (MITM): RSA được sử dụng để ký số và xác minh danh tính của các bên tham gia trao đổi khóa. Điều

này giúp đảm bảo rằng khóa được tạo ra là từ đúng đối tượng, tránh việc kẻ tấn công xen vào quá trình trao đổi.

Chuẩn hóa khóa cho DES bằng MD5: Sau khi khóa chung được tạo thành từ DH, hàm băm MD5 giúp chuẩn hóa về độ dài và cấu trúc phù hợp với yêu cầu của thuật toán DES. Dù MD5 không còn được sử dụng phổ biến cho bảo mật hiện đại, trong trường hợp này nó chỉ làm nhiệm vụ chuẩn hóa, nên vẫn đảm bảo hiệu quả và tốc độ xử lý.

Tạo ra mô hình mã hóa nhiều lớp: Việc phối hợp ba lớp bảo mật giúp tăng cường độ an toàn tổng thể. Nếu một lớp bị tấn công, các lớp còn lại vẫn giữ vai trò bảo vệ thông tin, từ đó nâng cao khả năng phòng thủ và hạn chế rủi ro.

Ứng dụng tốt trong môi trường mạng không an toàn: Mô hình tích hợp phù hợp với các hệ thống truyền thông số, hệ thống giám sát từ xa hoặc trao đổi thông tin quan trọng qua Internet, nơi mà yếu tố bảo mật được ưu tiên hàng đầu.

# CHƯƠNG 4: XÂY DỰNG ỨNG DỤNG WEB MINH HỌA

## THUẬT TOÁN DES

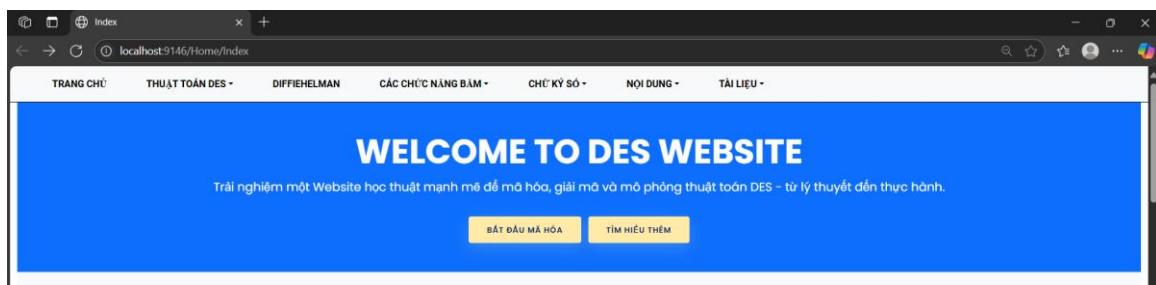
### 4.1. THIẾT KẾ GIAO DIỆN ỨNG DỤNG WEB

#### 4.1.1. Khối giới thiệu trang chủ:

Phần trang chủ là điểm đầu tiên người dùng tiếp cận khi truy cập ứng dụng web. Giao diện được thiết kế thân thiện, trực quan, sử dụng các tông màu nổi bật (Bootstrap theme) và chia bố cục rõ ràng theo từng vùng chức năng. Cụ thể như sau:

- Phần chào mừng nằm ở đầu trang với tông màu nền xanh dương (bg-primary), chứa tiêu đề nổi bật “WELCOME TO DES WEBSITE” và một đoạn mô tả ngắn giúp người dùng hiểu được mục tiêu của ứng dụng. Hai nút điều hướng nhanh giúp truy cập nhanh vào phần mã hóa hoặc hướng dẫn sử dụng.

Mục tiêu: Giới thiệu tổng quát, tạo ấn tượng mạnh mẽ và thúc đẩy người dùng ngay từ đầu.



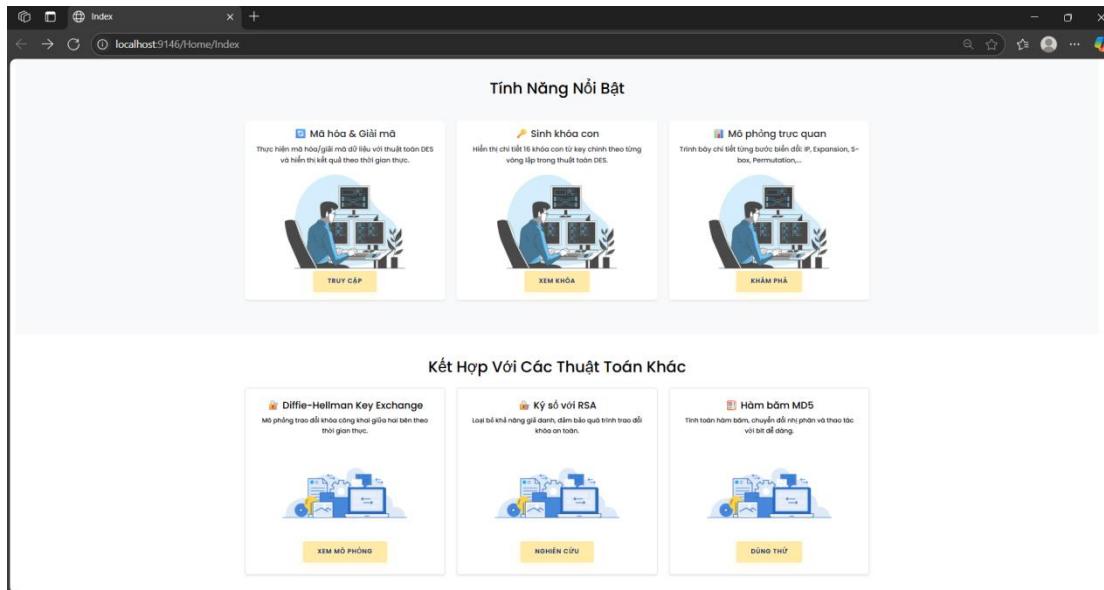
**Hình 4.1.** Giao diện trang chủ khi truy cập website

#### 4.1.2. Khối tính năng nổi bật và kết hợp với các thuật toán khác:

Phần tiếp theo là ba thẻ thông tin hiển thị các chức năng cốt lõi của hệ thống:

- Mã hóa & Giải mã: Cho phép người dùng nhập dữ liệu và thực hiện thuật toán DES.
- Sinh khóa con: Hiển thị 16 khóa con trong DES.
- Mô phỏng trực quan: Giải thích từng bước hoạt động của thuật toán (IP, S-box,...).

Mỗi tính năng đều có hình mô phỏng, mô tả ngắn và nút điều hướng tới trang liên quan.



**Hình 4.2.** Giao diện hiển thị tính năng hệ thống

Ứng dụng không chỉ tập trung vào DES, mà còn tích hợp các thuật toán khác để người học có cái nhìn toàn diện hơn:

- Diffie-Hellman: Mô phỏng trao đổi khóa.
- RSA: Ký số, xác thực người gửi.
- MD5: Băm và thao tác bit cơ bản.

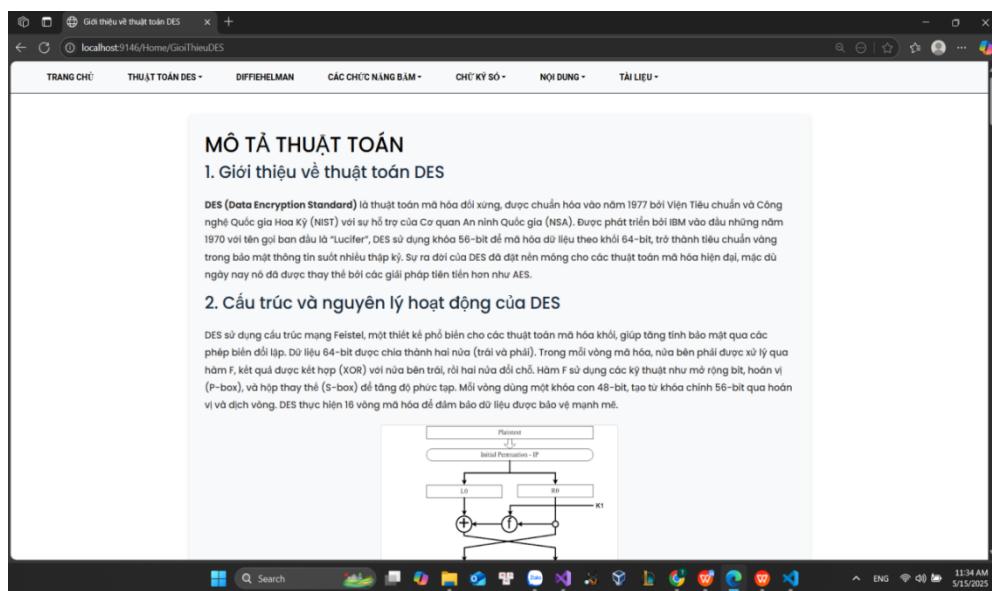
Mục này thể hiện rõ định hướng mở rộng và sự kết nối giữa các thuật toán mật mã hiện đại.

#### 4.1.3. Trang mô tả thuật toán và hướng dẫn sử dụng:

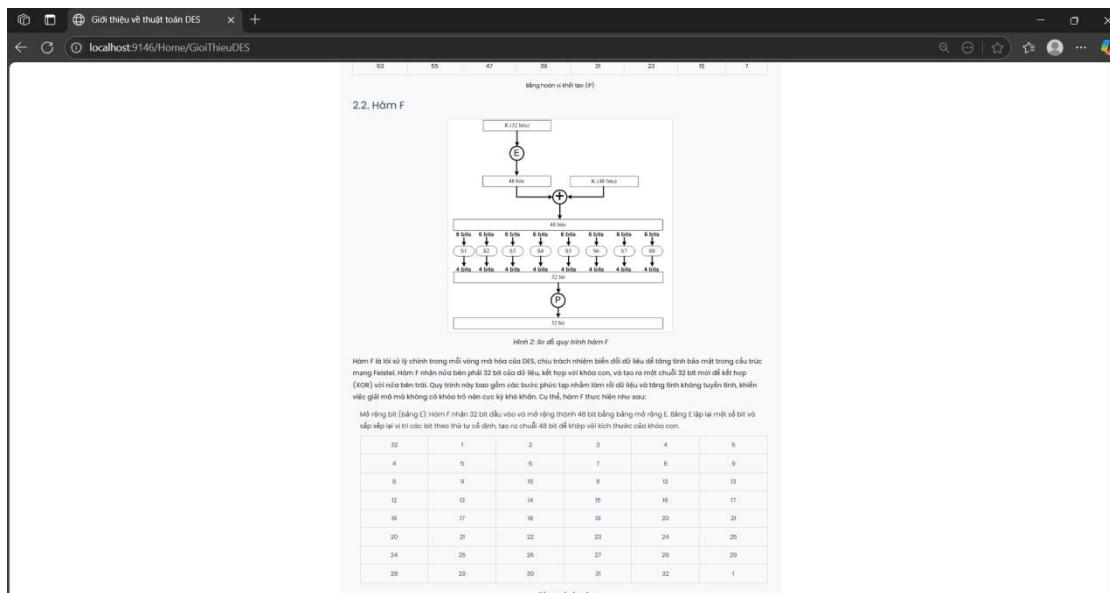
Ngoài giao diện trang chủ (Index), ứng dụng còn được thiết kế với các trang chức năng riêng biệt nhằm hỗ trợ người dùng trong việc tìm hiểu và sử dụng hệ thống. Trang này cung cấp nội dung lý thuyết và minh họa trực quan về thuật toán DES. Bao gồm:

- Lý thuyết và nguyên lý hoạt động của DES.
- Mô tả các bước trong quy trình mã hóa và giải mã.
- Hình ảnh minh họa luồng xử lý.
- Mô phỏng các khối chức năng chính như hoán vị ban đầu, chia khóa, vòng lặp mã hóa, hoán vị cuối.

Mục tiêu của trang này là giúp người dùng không chỉ sử dụng công cụ mà còn hiểu rõ cơ chế hoạt động của thuật toán DES.



**Hình 4.3.** Giao diện mô tả thuật toán DES 1



**Hình 4.4.** Giao diện mô tả thuật toán DES 2

Trang hướng dẫn được thiết kế nhằm giúp người dùng dễ dàng thao tác với website. Nội dung gồm:

- Hướng dẫn cách nhập văn bản cần mã hóa hoặc giải mã.
- Cách tạo và nhập khóa mã hóa.
- Cách thực hiện các thao tác mã hóa, giải mã, kiểm tra hàm băm.
- Kết hợp với các thuật toán khác

Mục tiêu của trang này đóng vai trò như tài liệu sử dụng giúp người dùng thao tác dễ dàng hơn và giải đáp thắc mắc. (Hình 5.5 và Hình 5.6).

**Hướng dẫn sử dụng công cụ mã phỏng mã hóa và giải mã DES**

Công cụ này mã phỏng quy trình mã hóa và giải mã dữ liệu bằng thuật toán DES (Data Encryption Standard). Giao thức Diffie-Hellman được sử dụng để trao đổi khóa an toàn, kết hợp với chữ ký số để xác minh danh tính, ngăn chặn tấn công man-in-the-middle (MITM). Người dùng nhập hai số bí mật (a và b) để tạo khóa chung, sau đó khóa này được băm bằng MD5 (lấy 16 ký tự HEX đầu tiên) để tạo khóa DES. Quy trình bao gồm:

- Bước 1: Trao đổi khóa công khai Diffie-Hellman.
- Bước 2: Tạo cặp khóa RSA cho chữ ký số.
- Bước 3: Xác minh danh tính người gửi qua chữ ký số.
- Bước 4: Tạo khóa chung bằng Diffie-Hellman.
- Bước 5: Băm khóa chung bằng MD5 để tạo khóa DES.
- Bước 6: Mã hóa hoặc giải mã dữ liệu bằng DES, hỗ trợ định dạng ASCII, HEX và BIN.

Lưu ý: Việc nhập cả hai số bí mật a và b chỉ dành cho mục đích mã phỏng. Trong thực tế, mỗi bên chỉ giữ số bí mật của mình và trao đổi khóa công khai qua kênh liên lạc.

**1. Tạo khóa với Diffie-Hellman**

Công cụ cho phép mã phỏng giao thức Diffie-Hellman bằng cách nhập cả hai số bí mật (a và b) để tính toán khóa công khai và khóa chung. Quy trình bao gồm:

- Nhập tham số
- Giá trị

**Hình 4.5.** Giao diện phần đầu hướng dẫn sử dụng web

**7. Minh họa toàn bộ quy trình**

Dưới đây là toàn bộ quy trình:

Thao tác	Đầu vào	Đầu ra
Tạo khóa Diffie-Hellman	$p = 23, q = 5$ Số bí mật: a = 18, b = 7	Khóa công khai: $a = 6, b = 17$ Khóa chung: c = 3
Tạo khóa RSA	Số nguyên tố: p = 61, q = 53 n = 37	Khóa công khai: $(e, n) = (37, 3233)$ Khóa riêng: $(d, n) = (2753, 3233)$
Tạo chữ ký số	Khóa bí mật (RSA) của A = 2753, dữ liệu cần ký là khóa công khai (diffie-hellman) của A = 6	$S = m^d \bmod n = 6^{2753} \bmod 3233 = 2982$
Xác minh chữ ký	Khóa công khai (RSA) của A = (17, 3233), chữ ký số S = 2982	$m = S^e \bmod n = 2982^{17} \bmod 3233 = 6 \rightarrow$ Xác minh thành công
Tạo khóa DES	Khóa chung: K = 3	Khóa DES: eccbc87e4b5e2f5e (16 ký tự HEX đầu của MD5)
Mã hóa	Bản rõ: Hello (ASCII) Khóa: eccbc87e4b5e2f5e (HEX)	Bản mã: 0845C0956A74B04A (HEX)
Giai mã	Bản mã: 0845C0956A74B04A (HEX) Khóa: eccbc87e4b5e2f5e (HEX)	Bản rõ: Hello (ASCII)

**8. Lưu ý quan trọng**

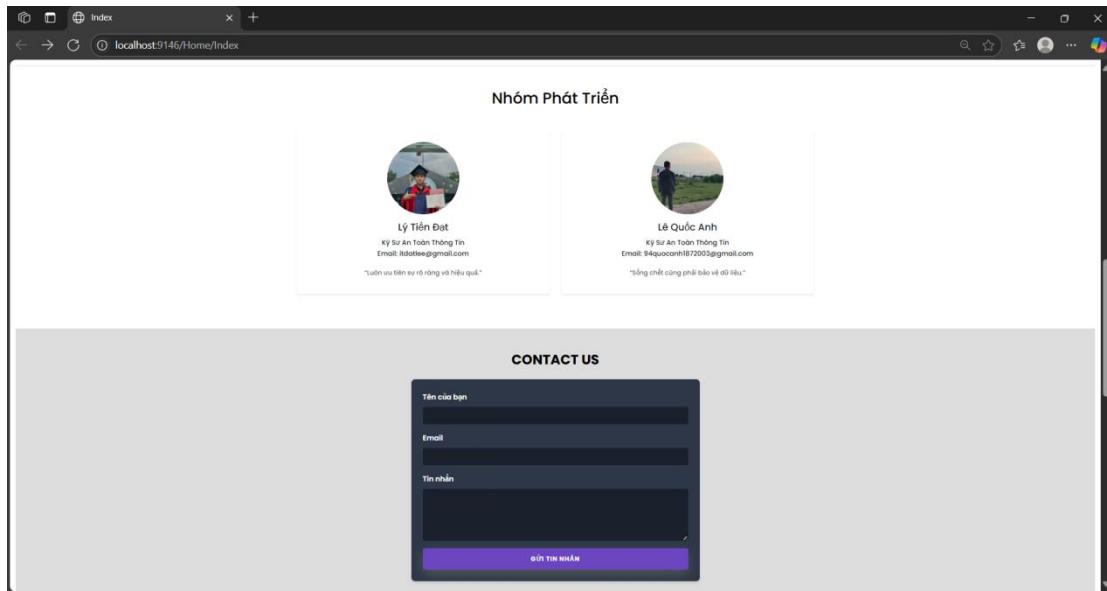
Đây là công cụ mã phỏng, cho phép nhập cả a và b để minh họa Diffie-Hellman. Trong thực tế, chỉ chia sẻ khóa công khai (a), không chia sẻ số bí mật. Khóa RSA phải được tạo với số nguyên tố lớn (2048 bit hoặc hơn) để đảm bảo an toàn trong thực tế.

**Hình 4.6.** Giao diện phần cuối hướng dẫn sử dụng web

#### 4.1.4. Thông tin nhóm phát triển trang web:

Hiển thị thành viên nhóm sáng lập ứng dụng với hình ảnh đại diện, họ tên, email liên hệ và câu nói truyền cảm hứng. Góp phần tăng tính minh bạch và thể hiện trách nhiệm của nhóm đối với sản phẩm.

Biểu mẫu liên hệ: Cho phép người dùng gửi câu hỏi hoặc góp ý cho nhà phát triển.

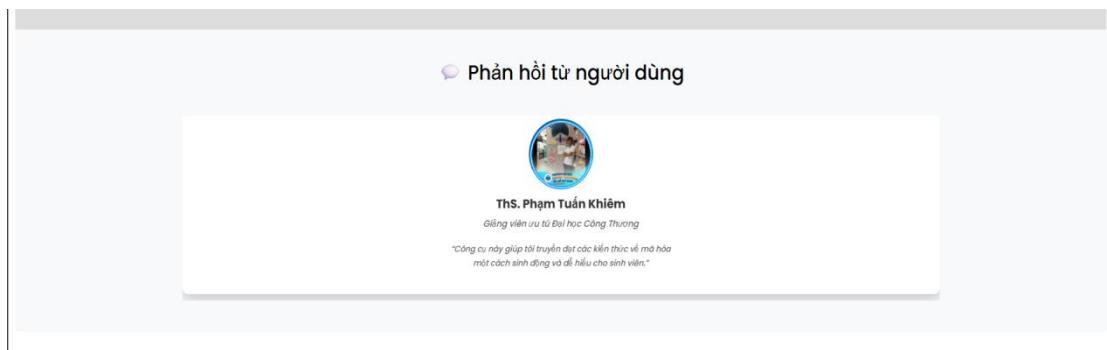


**Hình 4.7.** Thông tin nhà phát triển và liên hệ

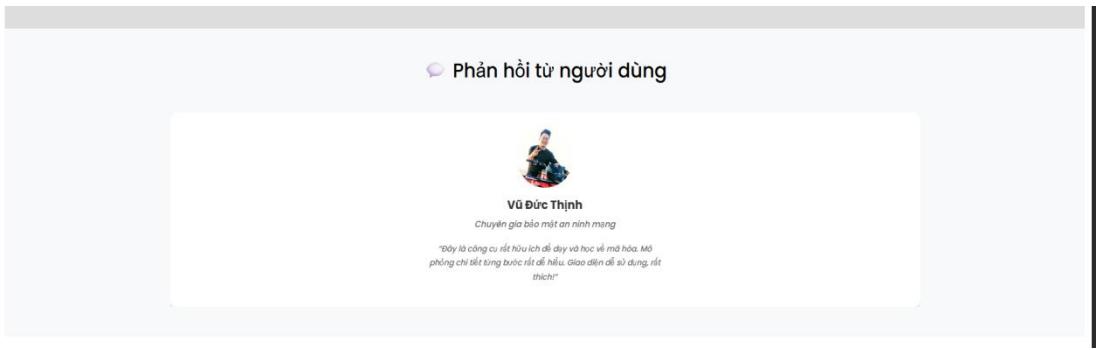
#### 4.1.5. Phản hồi của người dùng:

Phần cuối gồm:

- Phản hồi người dùng: Hiển thị đánh giá từ giảng viên và chuyên gia về tính ứng dụng và hiệu quả giảng dạy của phần mềm.



**Hình 4.8.** Phản hồi của người dùng 1



**Hình 4.9.** Phản hồi của người dùng 2

#### 4.1.6. Footer

Footer của website gồm có địa chỉ liên hệ và chủ quyền website của nhóm.



**Hình 4.10.** Footer chính của website

## 4.2. MÔ TẢ CHỨC NĂNG CỦA ỦNG DỤNG

Ứng dụng web được xây dựng nhằm minh họa quá trình mã hóa và giải mã dữ liệu sử dụng thuật toán DES, kết hợp với các kỹ thuật hỗ trợ để tăng cường tính bảo mật như tạo khóa chung, xác thực và băm dữ liệu. Các chức năng chính được triển khai bao gồm:

### 4.2.1. Tạo khóa chung bằng thuật toán Diffie-Hellman

Công cụ cho phép người dùng thực hiện trao đổi khóa thông qua thuật toán Diffie–Hellman (DH) bằng cách nhập cả hai số bí mật ( $a$  và  $b$ ) để tính toán khóa công khai và khóa chung. Quá trình này sử dụng các số nguyên tố và phép toán lũy thừa modulo để đảm bảo an toàn trước các cuộc tấn công nghe lén.

Bước 1: Nhập tham số Diffie-Hellman

Nhập số nguyên tố  $p$  và số nguyên  $g$  (generator). Công cụ có thể cung cấp giá trị mặc định.

Nhập số bí mật của hai bên:  $a$  và  $b$ . Ví dụ:

Tham số	Giá trị
$p$	23
$g$	5
Số bí mật $a$	18
Số bí mật $b$	7

**Bảng 4.1.** Ví dụ các tham số Diffie-Hellman

Mô phỏng thuật toán Diffie-Hellman

Số nguyên tố p:

Cơ số g (căn nguyên tố):

Alice chọn số bí mật a:

Bob chọn số bí mật b:

**SINH SỐ NGẪU NHÌN**    **TÍNH TOÁN**

**Hình 4.11.** Giao diện nhập tham số Diffie-Hellman trong hệ thống

Bước 2: Tính khóa công khai và khóa chung

Khóa công khai:

Khóa	Công thức	Ví dụ
A	$g^a \text{ mod } p$	$5^{18} \text{ mod } 23 = 6$
B	$g^b \text{ mod } p$	$5^7 \text{ mod } 23 = 17$

**Bảng 4.2.** Ví dụ cách tính khóa công khai Diffie-Hellman

## Khóa chung

Khóa	Công thức	Ví dụ
K	$A^b \text{ mod } p = B^a \text{ mod } p$	$17^{18} \text{ mod } 23 = 6^7 \text{ mod } 23 = 3$

**Bảng 4.3.** Ví dụ cách tính khóa chung Diffie-Hellman

**Hình 4.12.** Giao diện kết quả Diffie-Hellman trong hệ thống

**Lưu ý:** Trong mô phỏng, bạn nhập cả a và b để thấy toàn bộ quy trình từ tính khóa công khai đến khóa chung. Trong thực tế, b là bí mật của đối phương và không được chia sẻ, khóa công khai trước khi gửi cho người nhận cần phải được thực hiện cùng chữ ký số để xác minh.

### 4.2.2. Tạo cặp khóa RSA cho chữ ký số

Trước khi xác minh chữ ký số, cần tạo cặp khóa RSA gồm khóa công khai và khóa riêng để sử dụng trong quá trình ký và xác minh danh tính. Khóa riêng được giữ bí mật, còn khóa công khai được chia sẻ để xác minh chữ ký.

Bước 1: Tạo cặp khóa RSA

Chọn hai số nguyên tố lớn  $p$  và  $q$  (ví dụ:  $p = 61$ ,  $q = 53$ ).

Tính  $n = p * q$  và hàm Euler  $\varphi(n) = (p-1) * (q-1)$ .

Chọn số nguyên  $e$  sao cho  $1 < e < \varphi(n)$  và  $e$  nguyên tố cùng nhau với  $\varphi(n)$ .

Tính  $d$  sao cho  $d * e \equiv 1 \pmod{\varphi(n)}$ .

Kết quả:

Tham số	Ví dụ
$n$	$61 * 53 = 3233$
$\varphi(n)$	$(61-1) * (53-1) = 3120$
Khóa công khai	$(e, n) = (17, 3233)$
Khóa riêng	$(d, n) = (2753, 3233)$

**Bảng 4.4.** Ví dụ tạo cặp khóa RSA

Bước 2: Lưu và chia sẻ khóa

Lưu khóa riêng  $(d, n)$  ở nơi an toàn, không chia sẻ.

Chia sẻ khóa công khai  $(e, n)$  với đối phương để họ sử dụng trong xác minh chữ ký.

Bước 3: Nhấn nút "Tạo khóa RSA"

Công cụ sẽ hiển thị cặp khóa RSA  $(e, n)$  và  $d, n$ .

**Hình 4.13.** Giao diện tạo cặp khóa RSA

**Lưu ý:** Trong thực tế, các số nguyên tố p và q phải rất lớn (thường 2048 bit hoặc hơn) để đảm bảo an toàn. Trong mô phỏng, các số nhỏ được dùng để đơn giản hóa.

#### 4.2.3. Xác thực khóa bằng chữ ký số RSA

Để đảm bảo khóa công khai từ A gửi đến B đáng tin cậy, công cụ sử dụng chữ ký số để xác minh danh tính.

Bước 1: Nhận chữ ký số

Người gửi (A) sẽ gửi khóa công khai kèm chữ ký số (tạo bằng thuật toán RSA với khóa riêng) cho người nhận (B).

Chữ ký số (S) là 1 giá trị được tạo bằng cách ký thông điệp (trong ví dụ này thông điệp là khóa công khai của giao thức Diffie Hellman ở phần 1)

**RSA - Ký số**

Khóa bí mật d:	Môđun n:
2753	3233
Dữ liệu cần ký (m):	
6	
<b>THỰC HIỆN</b>	

**Kết quả**

Khóa bí mật (d, n): (2753, 3233)  
Dữ liệu gốc (m): 6  
Chữ ký (s): 2902  
Quá trình: Ký số: s = m^d mod n = 6^2753 mod 3233 = 2902

### Hình 4.14. Giao diện ký số RSA

Kết quả nhận được:  $S = m^d \text{ mod } n = 6^{2753} \text{ mod } 3233 = 2902$  (đây là chữ ký). Người gửi (A) sẽ gửi khóa công khai Diffie-Hellman và chữ ký cho (B): (6,2902)

#### Bước 2: Nhập thông tin xác minh

Người nhận B sau khi nhận được thông điệp (khóa công khai Diffie-Hellman) và chữ ký từ A (6,2902) sẽ kiểm xác minh bằng khóa công khai của A.

Ví dụ:

Tham số	Giá trị
Thông điệp cần xác minh	6
Chữ ký số	2902
Khóa công khai RSA	(e, n) = (17, 3233)

**Bảng 4.5.** Ví dụ các tham số tạo chữ ký

### Bước 3: Thực hiện kiểm tra

RSA – Kiểm tra chữ ký

Khóa công khai e:	Mô-đun n:
17	3233
Dữ liệu cần kiểm tra (s):	2902
<b>THỰC HIỆN</b>	

Kết quả

Khóa công khai (e, n): (17, 3233)  
Chữ ký (s): 2902  
Dữ liệu kiểm tra (m): 6  
Quá trình: Kiểm tra:  $m' = s^e \bmod n = 2902^{17} \bmod 3233 = 6$

**Hình 4.15.** Giao diện xác minh danh tính RSA

Kết quả nhận được:  $m = s^e \bmod n = 2902^{17} \bmod 3233 = 6$  (kết quả sau khi xác minh chữ ký trùng khớp với nội dung mà A gửi)  $\rightarrow$  thông điệp không bị thay đổi

**Lưu ý:** Nếu xác minh thất bại (kết quả khi xác minh chữ ký không trùng khớp với thông điệp), thì không sử dụng vì có thể thông điệp đã bị tấn công MITM.

#### 4.2.4. Băm khóa bằng MD5 và trích xuất khóa DES

Sau khi xác minh B, công cụ băm khóa chung bằng MD5 để tạo khóa DES.

### Bước 1: Khóa chung

Khóa chung K đã được tính ở bước 1 ( $K = 3$ ).

**Băm khóa chung bằng MD5**

Nhập khóa chung:

**BẤM KHÓA**

**Hình 4.16.** Băm khóa với MD5

### Bước 2: Băm khóa chung bằng MD5

Công cụ băm K bằng MD5 băm ra được 128 bit. Kết quả băm, hay còn gọi là hash, của MD5 có độ dài 128 bit. Điều này tương đương với 16 byte

**Băm khóa chung bằng MD5**

Nhập khóa chung:

**BẤM KHÓA**

**Kết quả:**

**Khóa chung đã nhập:**

**Khóa sau khi băm MD5:**

**Hình 4.17.** Kết quả băm khóa với MD5

### Bước 3: Trích xuất 64 bit đầu từ MD5

Vì thuật toán DES cần dữ liệu đầu vào có độ dài 64 bit tương ứng 8 byte) lấy 16 ký tự HEX đầu tiên (8 byte) làm khóa DES.

**Trích xuất 64-bit đầu từ MD5**

Nhập chuỗi MD5:

**LẤY 64-BIT ĐẦU TIÊN**

**Kết quả:**

**Chuỗi MD5 ban đầu:**

**64-bit đầu tiên (16 ký tự hex):**

**Hình 4.18.** Trích xuất từ MD5

Bước 4: Chuyển đổi về nhị phân

Ta chuyển đổi từ HEX về BIN

**Chuyển đổi Hex sang Nhị phân**

Nhập 16 ký tự hex:

**CHUYỂN ĐỔI**

**Kết quả:**

**Chuỗi hex ban đầu:**

**Dạng nhị phân (64-bit):**

**Hình 4.19.** Chuyển đổi định dạng khóa

Bin là dạng nhị phân chỉ dùng 0 và 1.

Ta được

Thao tác	Giá trị
Khóa chung K	3
MD5(K)	eccbc87e4b5ce2fe28308fd9f2a7baf3
Khóa DES (HEX)	eccbc87e4b5ce2fe
Khóa DES (BIN)	111011001100101110010000111110 0100101101011100111000101111110

**Bảng 4.6.** Ví dụ băm khóa và chuyển đổi

#### 4.2.5. Mã hóa dữ liệu với DES

Sử dụng khóa DES để mã hóa thông điệp. Quy trình bao gồm:

Bước 1: Nhập thông điệp (Plaintext)

Chọn định dạng: ASCII, HEX, hoặc BIN. Ví dụ:

Định dạng	Giá trị
ASCII	Là dạng chữ (Chỉ hỗ trợ tiếng Anh) như "Hello".
HEX	Là dạng mã hóa thành số kiểu hệ 16, từ 0-9 và A-F. Ví dụ: eccbc87e4b5ce2fe
BIN	Là dạng chỉ dùng 0 và 1. Ví dụ: 1110110011001011100100001111100 0100101101011100111000101111110

**Bảng 4.7.** Các dạng dữ liệu đầu vào

## Bước 2: Nhập khóa DES

Chọn định dạng: ASCII, HEX, hoặc BIN.

Nhập khóa DES từ bài trước eccbc87e4b5ce2fe khóa này ở dạng HEX.

Bước 3: Nhấn nút "Xử lý dữ liệu"

**Hình 4.20.** Mã hóa thuật toán DES

**Lưu ý:** Thông điệp nếu dài hơn 64 bit sẽ được chia thành các khối 64 bit (các khối không đủ sẽ được padding). Kết quả sau khi mã hóa sẽ ở dạng Hex hoặc Bin, dùng dạng nào để giải mã cũng được.

#### **4.2.6. Giải mã dữ liệu với DES**

Sử dụng khóa DES để giải mã bản mã. Quy trình bao gồm:

#### Bước 1: Nhập bản mã (Ciphertext)

**Chọn định dạng: HEX hoặc BIN.**

D845C0056A740A0A (HEX).

## Bước 2: Nhập khóa DES

Nhập khóa DES giống bước mã hóa eccbc87e4b5ce2fe.

### Bước 3: Nhấn nút "Xử lý dữ liệu"

### Hình 4.21. Giải mã thuật toán DES

Dữ liệu đã được giải mã về thông điệp ban đầu.

Đầu vào	Giá trị
Thông điệp cần giải mã	D845C0056A740A0A (Hex).
Khóa	eccbc87e4b5ce2fe (Hex)
Kết quả	48656C6C6F000000 (Hex)

#### Bảng 4.8. Tổng kết giải mã

#### 4.2.7. Tổng kết toàn bộ quy trình minh họa:

Thao tác	Đầu vào	Đầu ra
Tạo khóa Diffie-Hellman	$p = 23, g = 5$ Số bí mật: $a = 18, b = 7$	Khóa công khai: $A = 6, B = 17$ Khóa chung: $K = 3$
Tạo khóa RSA	Số nguyên tố: $p = 61, q = 53$ $e = 17$	Khóa công khai: $(e, n) = (17, 3233)$ Khóa riêng: $(d, n) = (2753, 3233)$
Tạo chữ ký số	Khóa bí mật (RSA) của $A = 2753$ , dữ liệu cần ký là khóa công khai (Diffie-Hellman) của $A = 6$	$S = m^d \text{ mod } n = 6^{2753} \text{ mod } 3233 = 2902$
Xác minh chữ ký	Khóa công khai (RSA) của $A = (17, 3233)$ , chữ ký $S = 2902$	$m = s^e \text{ mod } n = 2902^{17} \text{ mod } 3233 = 6 \rightarrow$ Xác minh đúng
Tạo khóa DES	Khóa chung: $K = 3$	Khóa DES: eccbc87e4b5ce2fe (16 ký tự HEX đầu của MD5)
Mã hóa	Bản rõ: Hello (ASCII) Khóa: eccbc87e4b5ce2fe (HEX)	Bản mã: D845C0056A740A0A (HEX)
Giải mã	Bản mã: D845C0056A740A0A (HEX) Khóa: eccbc87e4b5ce2fe (HEX)	Bản rõ: Hello (ASCII)

**Bảng 4.9.** Toàn bộ quy trình

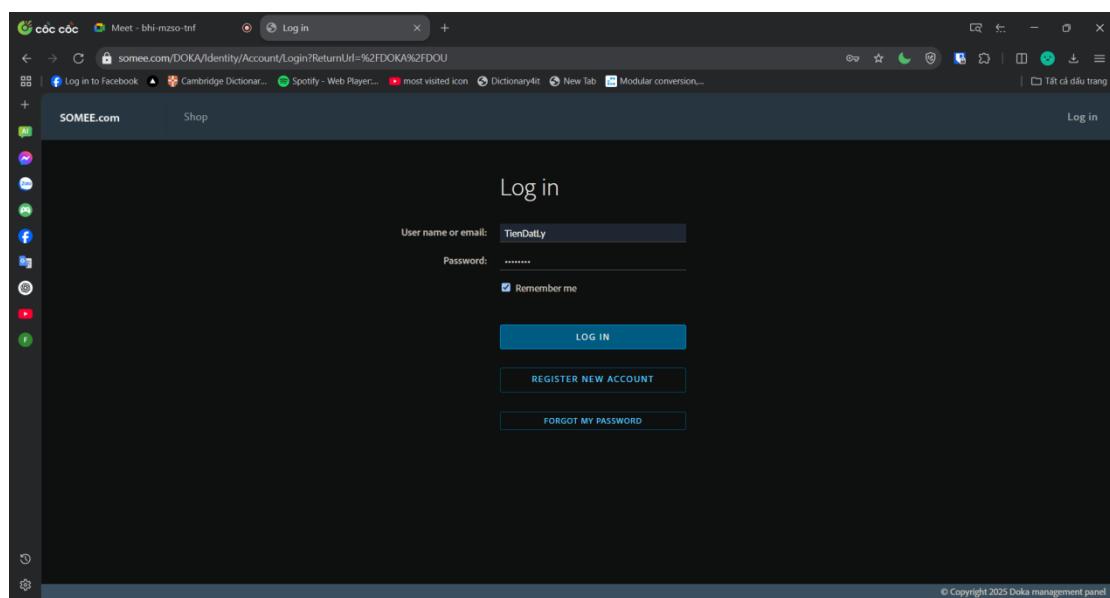
### 4.3. TRIỂN KHAI VÀ XUẤT BẢN WEB LÊN INTERNET

Sau khi hoàn thiện ứng dụng minh họa thuật toán mã hóa DES, nhóm em tiến hành triển khai và xuất bản website lên Internet nhằm phục vụ cho việc sử dụng từ xa áp dụng vào thực tế. Việc triển khai được thực hiện thông qua dịch vụ hosting miễn phí Somee.com, hỗ trợ ASP.NET phù hợp với dự án đang sử dụng.

Các bước thực hiện triển khai như sau:

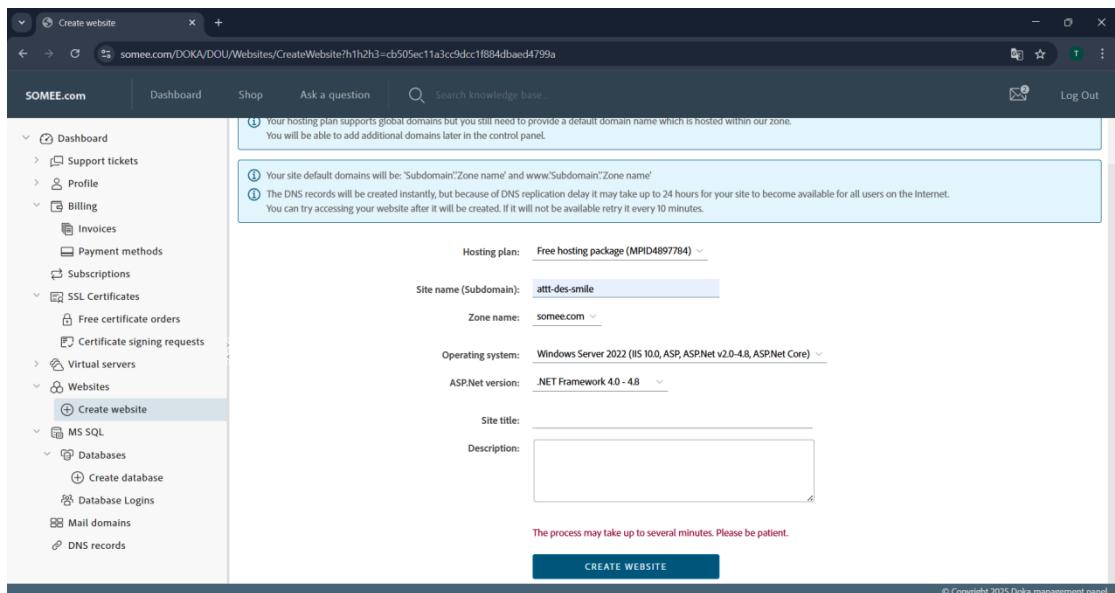
Bước 1: Tạo website trên Somee.com

Nhóm tiến hành đăng ký tài khoản tại địa chỉ <https://somee.com>, rồi đăng nhập vào.



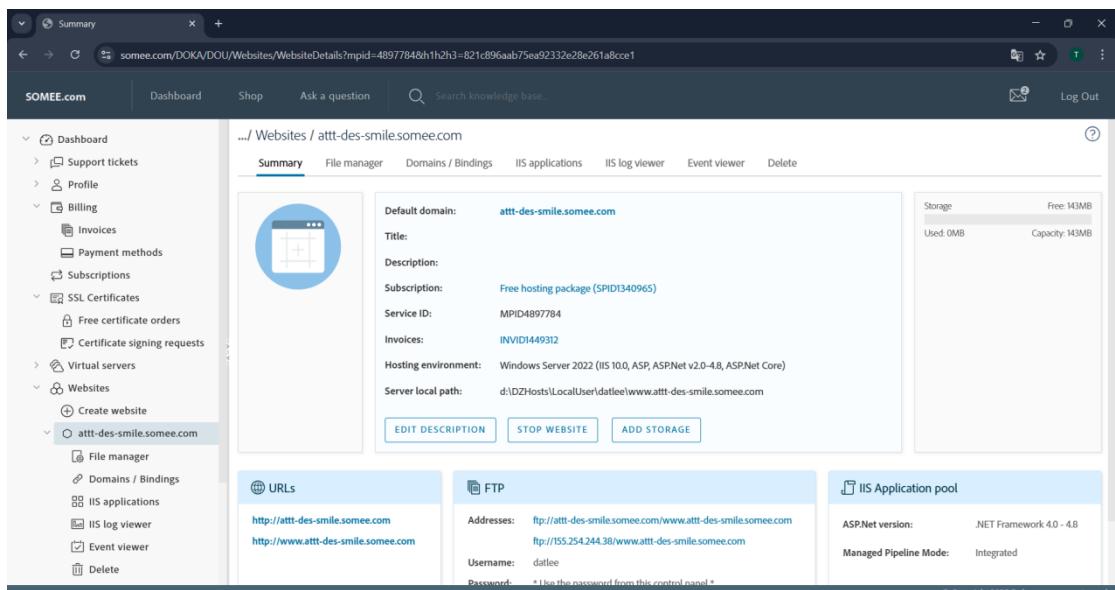
**Hình 4.22.** Trang đăng nhập của Somee

Sau đó tạo mới một website ASP.NET miễn phí. Ở đây nhóm em sẽ đặt domain website là attt-des-smile.

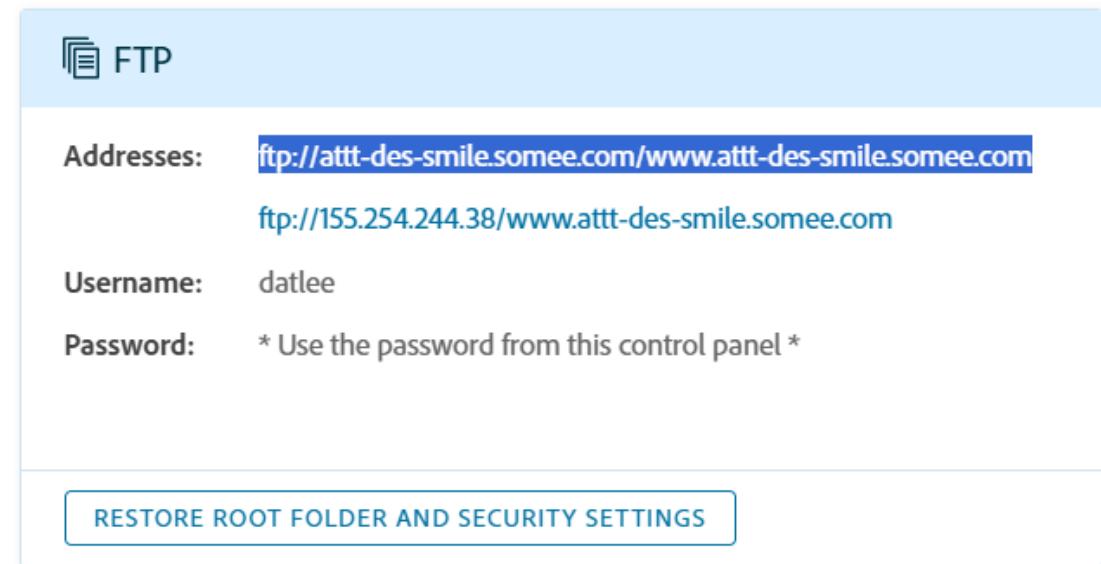


**Hình 4.23.** Đặt tên domain cho website

Hệ thống sẽ cung cấp thông tin FTP và đường dẫn truy cập website.



**Hình 4.24.** Giao diện khi tạo xong website

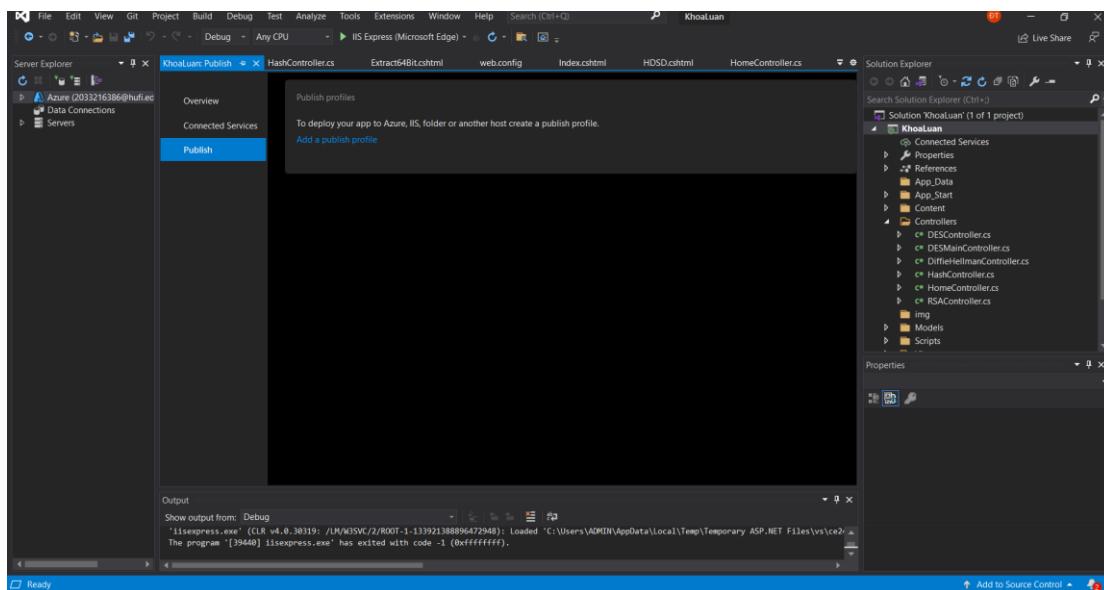


**Hình 4.25.** Thông tin FTP và link website

Copy đoạn FTP này để chuẩn bị cho việc thực hiện thao tác tiếp theo.

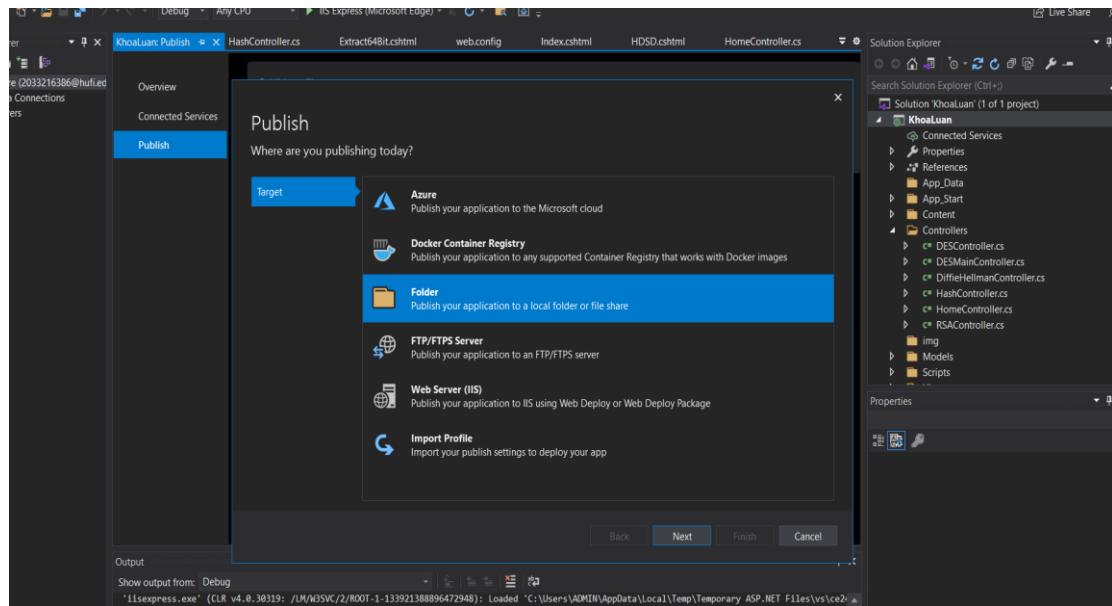
Bước 2: Vào Visual Studio để publish sản phẩm code.

Chuột phải vào KhoaLuan chọn publish.



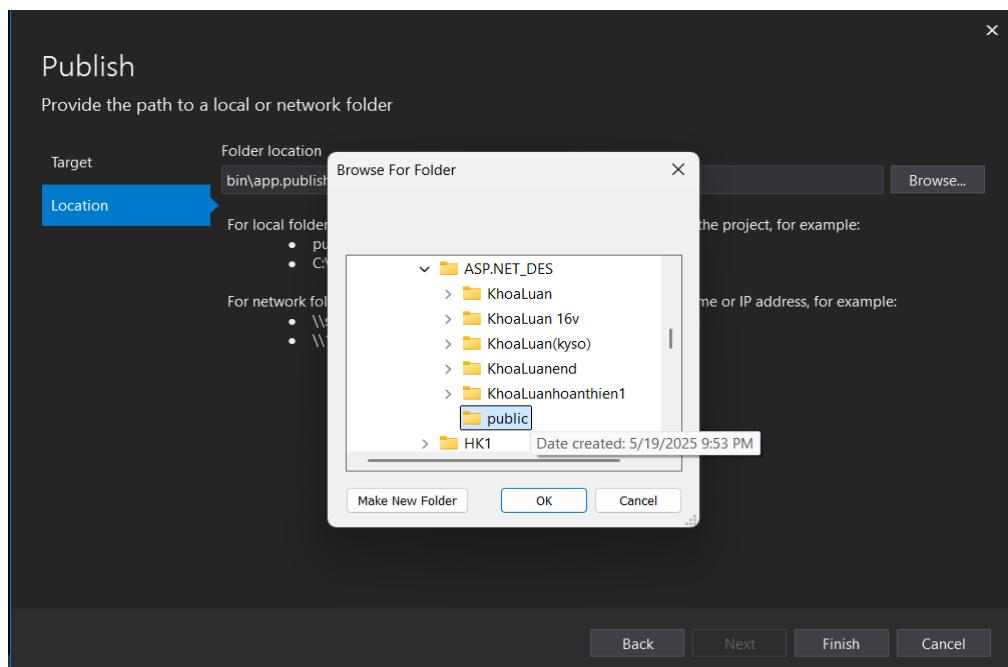
**Hình 4.26.** Giao diện khi nhập publish

Chọn Folder -> Next



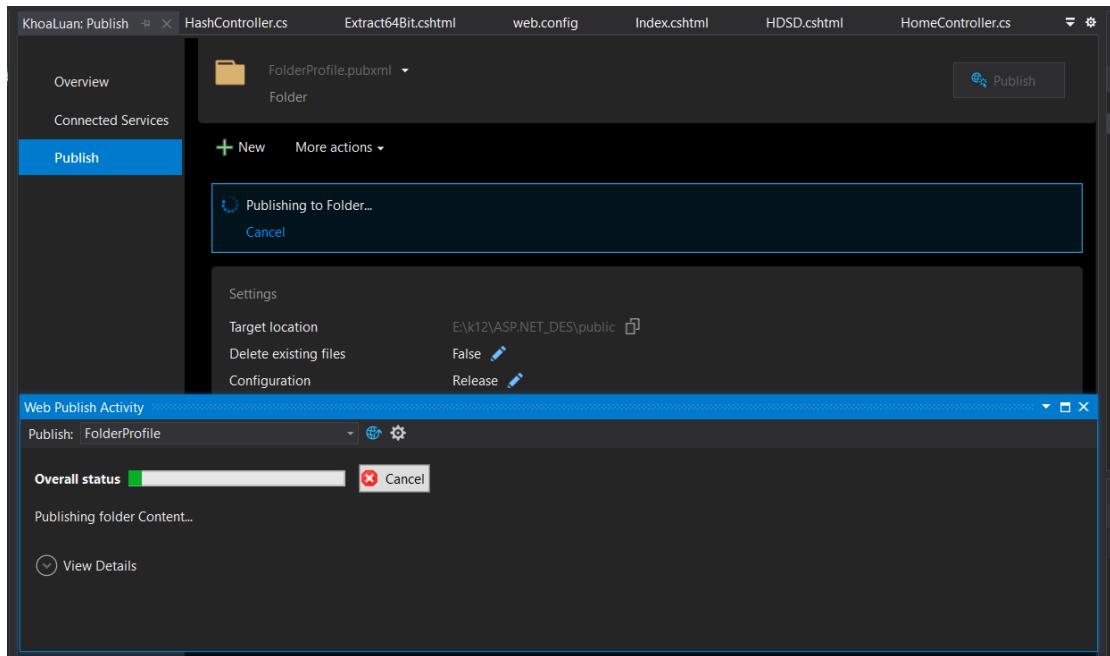
Hình 4.27. Publish folder

Chọn đường dẫn lưu trữ file để chia sẻ file sang Somee.

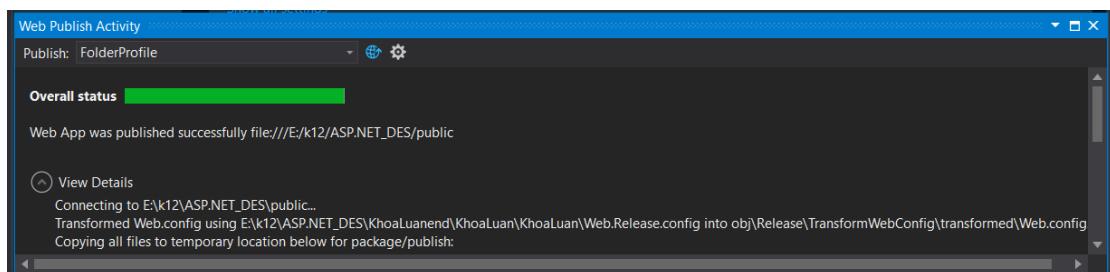


Hình 4.28. Đường dẫn lưu trữ

Chọn thành công, nhấp vào Publish để tiếp tục



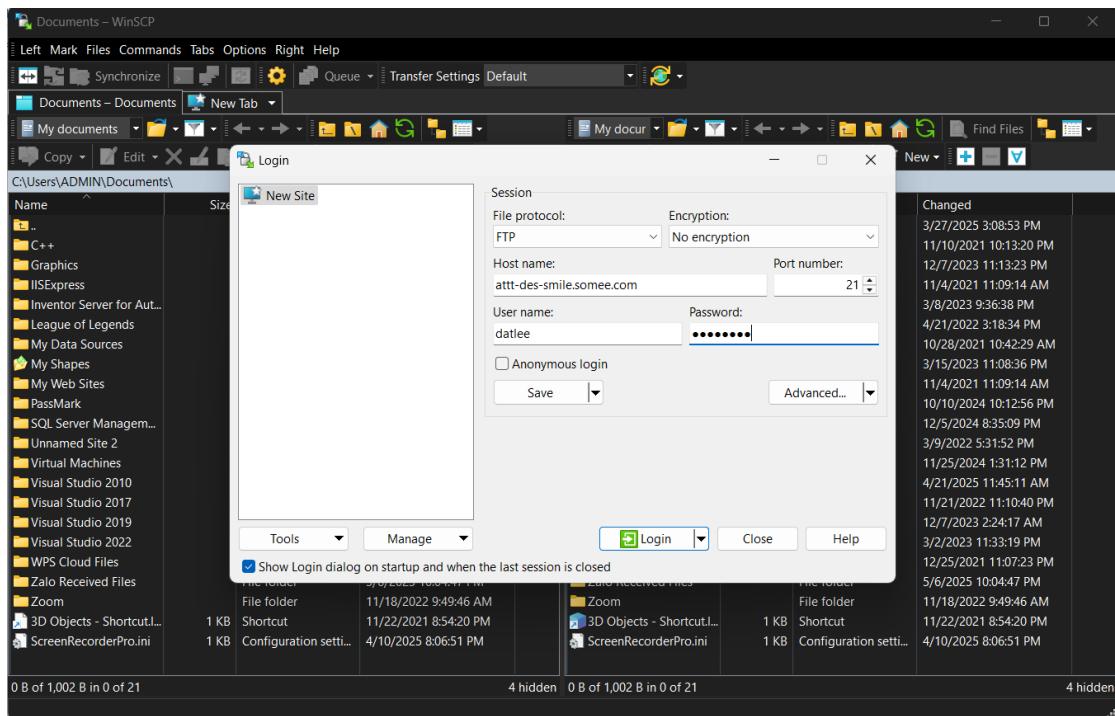
**Hình 4.29.** Quá trình publish đang thực hiện



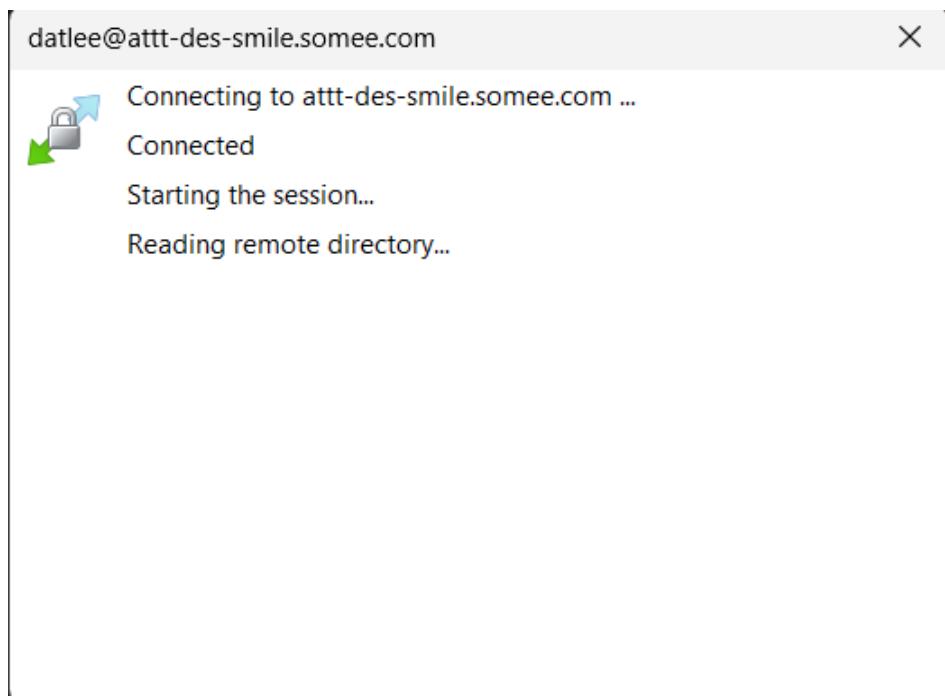
**Hình 4.30.** Quá trình publish thành công

### Bước 3: Đăng nhập và upload bằng WinSCP

Nhóm em sử dụng phần mềm WinSCP để đăng nhập vào server FTP mà Somee cung cấp. Sau đó tiến hành upload toàn bộ mã nguồn đã publish lên thư mục wwwroot trên hosting.

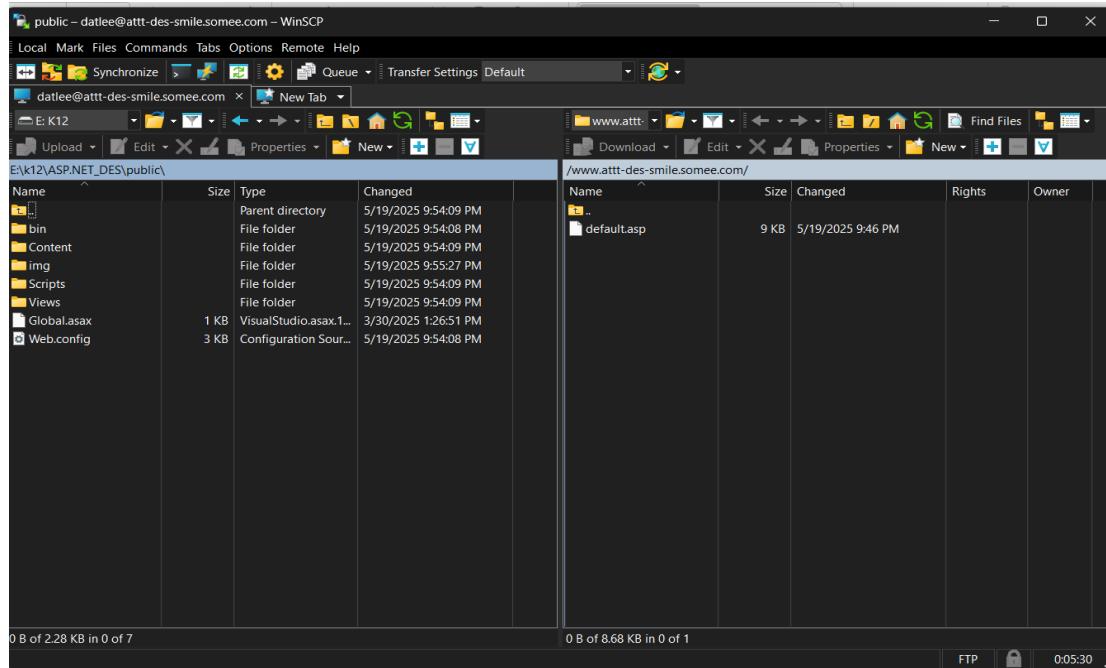


**Hình 4.31.** Giao diện login vào WinSCP



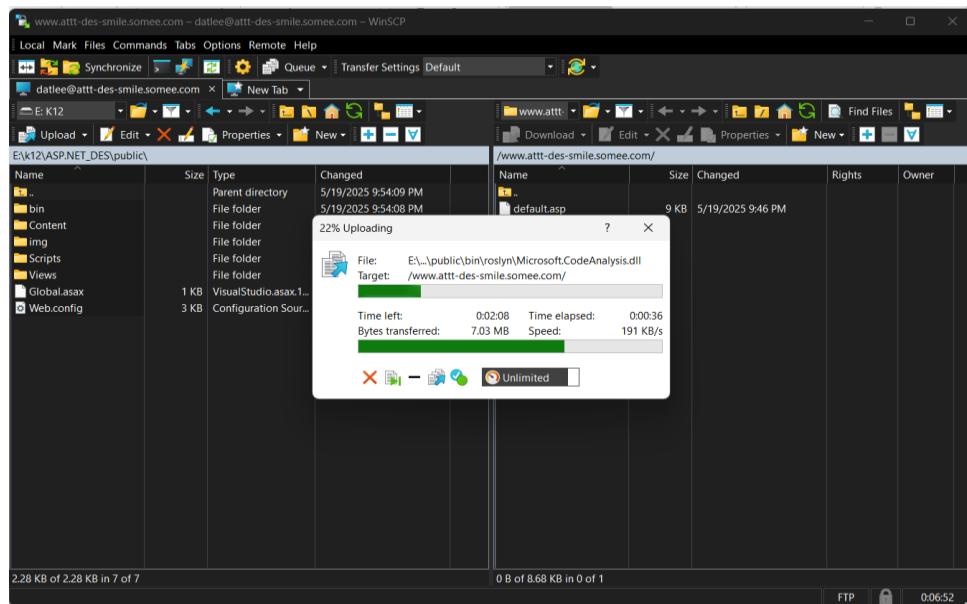
**Hình 4.32.** Giao diện chờ kết nối WinSCP

## Kết nối thành công giao diện thao tác sẽ hiện ra



**Hình 4.33.** Giao diện thao tác sau khi kết nối

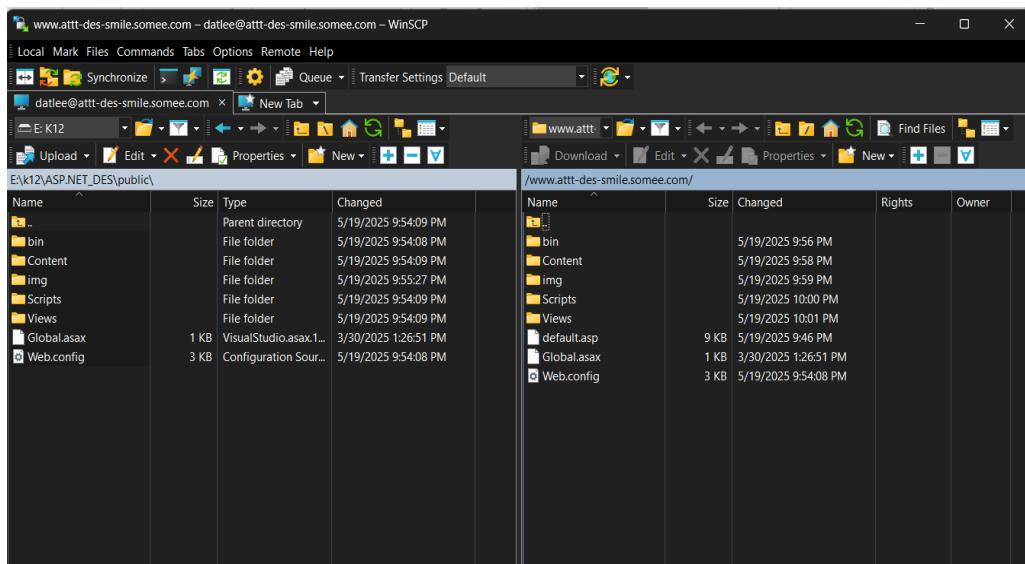
Sau đó copy các file của source code qua sang Somee.



**Hình 4.34.** Quá trình upload file code

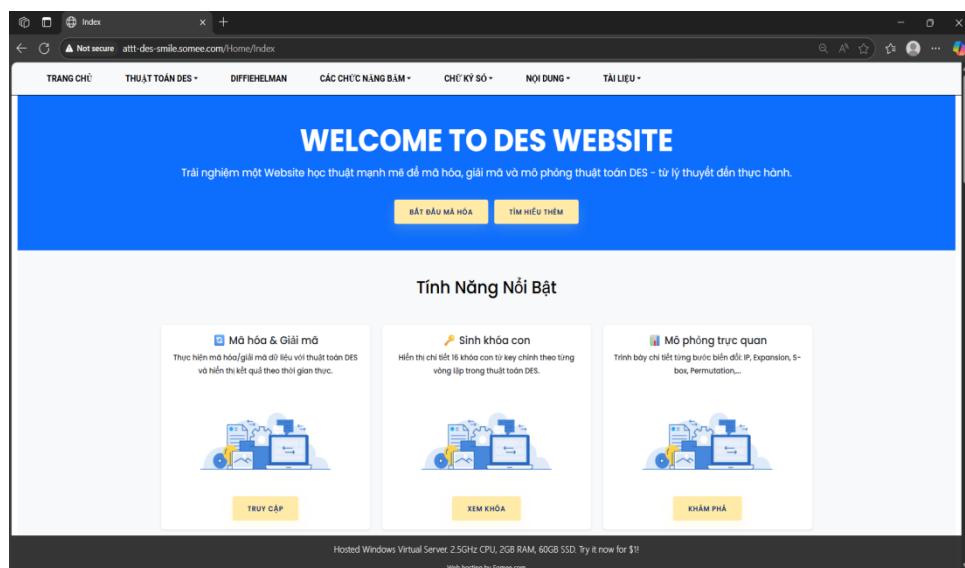
## Bước 4: Kiểm tra hoạt động website

Sau khi upload xong, truy cập website thông qua đường dẫn mà mình đã đặt do Somee cung cấp ở đây là <http://attt-des-smile.somee.com/>



**Hình 4.35.** Upload file code xong Somee thành công

Tiến hành vào để kiểm tra hoạt động và toàn bộ tính năng của hệ thống.



**Hình 4.36.** Giao diện website đã được public lên Internet

## KẾT LUẬN

Qua quá trình nghiên cứu và triển khai đề tài "Tìm hiểu thuật toán mã hóa DES và xây dựng ứng dụng minh họa chi tiết quá trình mã hóa và giải mã", nhóm chúng em đã nắm rõ cơ chế hoạt động của thuật toán mã hóa đối xứng DES, đồng thời nhận thức được những hạn chế về mặt bảo mật khi sử dụng một khóa chung cố định. Để nâng cao tính an toàn trong truyền thông số, nhóm em đã có kết hợp sử dụng thuật toán trao đổi khóa Diffie–Hellman để tạo khóa đầu vào cho DES, sau đó băm bằng MD5 để định dạng lại khóa. Tuy nhiên, nhận thấy DH có nguy cơ bị tấn công trung gian (MITM), nhóm em tiếp tục tích hợp thuật toán mã hóa bắt đối xứng RSA để thực hiện chức năng xác minh bằng chữ ký số, nhằm xác minh tính xác thực của bên tham gia trao đổi khóa. Sự kết hợp này đã tạo nên một hệ thống mã hóa gồm ba lớp bảo mật, giúp tăng cường tính toàn vẹn và bảo mật dữ liệu trong quá trình truyền tải. Ứng dụng minh họa được xây dựng không chỉ mô phỏng trực quan các bước trong quá trình mã hóa – giải mã, mà còn thể hiện sự phối hợp giữa các thuật toán trong việc đảm bảo an toàn thông tin. Cuối cùng là triển khai ứng dụng lên nền tảng Internet thông qua dịch vụ Somee, hỗ trợ truy cập từ xa để sử dụng và đánh giá hiệu suất. Nhóm chúng em rất cảm kích và biết ơn khoa vì thầy cô đã cho nhóm em nghiên cứu một đề tài rất hay và ý nghĩa. Đây sẽ là nền tảng quan trọng để chúng em tiếp tục nghiên cứu sâu hơn về các giải pháp bảo mật trong lĩnh vực mật mã học và an toàn thông tin.

## TÀI LIỆU THAM KHẢO

- PGS.TS Dương Anh Đức, Trần Minh Triết, *Giáo trình Mật mã học*, Trường Đại Học Công Nghệ Thông Tin - Đại Học Quốc Gia Thành Phố Chí Minh.
- Trần Minh Triết (2004), *Nghiên cứu một số vấn đề về bảo vệ thông tin và ứng dụng*, Luận văn Thạc sĩ Tin học, Đại học Khoa học Tự nhiên, Đại học Quốc gia thành phố Hồ Chí Minh.
- H. Feistel (1973), *Cryptography and computer privacy*, Scientific American, Vol. 228, No. 5, pp. 15-23.
- E.Biham, A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.
- E.Biham, A Shamir (1991), *Differential Cryptanalysis of DES-like cryptosystems*, Journal of Cryptology, Vol. 4, No. 1, pp.3-72
- F.Pub, *Data Encryption Standard (DES)*, National Institute of Standards and Technology (NIST), 1999
- Đinh Nguyễn Trọng Nghĩa (2022), *ASP.NET MVC Toàn tập - Lập trình Web HIFI* [Video playlist]. Youtube.  
<https://www.youtube.com/playlist?list=PLf5IPckgFwFUdtFXnvNjwgFdflTjKk0gF>
- William Stallings, *COMPUTER SECURITY: PRINCIPLES AND PRACTICE*, United Kingdom : Pearson, 2018.
- William Stallings, *Cryptography and Network Security Principles and Practice*, New York : Pearson, 2023.