**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🙜🙢🙠🙞**



**ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI: Tìm hiểu công nghệ Jamstack và xây dựng ứng dụng Web minh họa, bài toán khóa mở rộng trong thuật toán AES**

Giảng viên hướng dẫn: Phạm Tuấn Khiêm

Sinh viên thực hiện:

1. 2033210637 – Nguyễn Trung Kha

2. 2033212035 – Nguyễn Việt Thành

3. 2033216442 – Bồ Quang Khải

Thành phố Hồ Chí Minh – 2025

**LỜI CAM ĐOAN**

Tôi xin cam đoan: Đề tài “Nêu tên đề tài” là công trình nghiên cứu của cá nhân tôi dưới sự hướng dẫn của Ths Phạm Tuấn Khiêm. Các số liệu, kết quả nêu trong Đồ án là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác

Tôi xin chịu hoàn toàn trách nhiệm về lời cam đoan của mình.

**Sinh viên thực hiện Đồ án**

Bồ Quang Khải

Nguyễn Việt Thành

Nguyễn Trung Kha

**LỜI CẢM ƠN**

Quá trình thực hiện đồ án chuyên ngành là một giai đoạn quan trọng và đầy thử thách đối với mỗi sinh viên, bởi đây là kết quả của quá trình tích lũy kiến thức và kỹ năng trong suốt 4 năm học tập tại trường. Đồ án không chỉ là cơ hội để chúng em áp dụng những kiến thức đã học vào thực tế, mà còn giúp rèn luyện kỹ năng nghiên cứu, giải quyết vấn đề và chuẩn bị hành trang cho con đường nghề nghiệp phía trước.

Chúng em xin bày tỏ lòng biết ơn sâu sắc tới Ban Giám Hiệu cùng các thầy cô trong khoa Công nghệ Thông tin, Trường Đại học Công Thương TP. Hồ Chí Minh. Nhờ sự chỉ dẫn tận tình và sự hỗ trợ quý báu của các thầy cô, chúng em mới có thể hoàn thành đồ án này một cách tốt nhất.

Đặc biệt, chúng em xin gửi lời cảm ơn chân thành tới Thầy Phạm Tuấn Khiêm, người đã trực tiếp hướng dẫn và luôn sẵn sàng dành thời gian, tâm huyết để giúp đỡ, động viên chúng em trong suốt quá trình thực hiện đồ án.

Cuối cùng, chúng em xin gửi lời tri ân đến gia đình, bạn bè và những người thân yêu đã luôn đồng hành, động viên và hỗ trợ chúng em trong suốt hành trình này.

Chúng em xin chân thành cảm ơn!

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

Tên đề tài: “Tìm hiểu công nghệ Jamstack và xây dựng ứng dụng Web minh họa, bài toán khóa mở rộng trong thuật toán AES”

Sinh viên thực hiện:

2033210637 – Nguyễn Trung Kha

2033212035 – Nguyễn Việt Thành

2033216442 – Bồ Quang Khải

Khoá: 12DHBM

Nội dung nhận xét ........................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................ ........................................................................................................................................................... ...........................................................................................................................................................

Ngày …… tháng ……. năm 2024

Giảng viên hướng dẫn

**NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN**

Tên đề tài: “Tìm hiểu công nghệ Jamstack và xây dựng ứng dụng Web minh họa, bài toán khóa mở rộng trong thuật toán AES”

Sinh viên thực hiện:

2033210637 – Nguyễn Trung Kha

2033212035 – Nguyễn Việt Thành

2033216442 – Bồ Quang Khải

Khoá: 12DHBM

Nội dung nhận xét ........................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................ ........................................................................................................................................................... ...........................................................................................................................................................

Ngày …… tháng ……. năm 2024

Giảng viên phản biện

**LỜI MỞ ĐẦU**

Trong kỷ nguyên số hiện nay, an toàn thông tin và hiệu suất phát triển web là hai yếu tố then chốt trong việc xây dựng các hệ thống ứng dụng hiện đại. Một mặt, nhu cầu bảo vệ dữ liệu cá nhân và thông tin nhạy cảm ngày càng trở nên cấp thiết, kéo theo sự phát triển mạnh mẽ của các thuật toán mã hóa như AES (Advanced Encryption Standard). Trong đó, quá trình khóa mở rộng đóng vai trò cốt lõi, quyết định tính bảo mật của toàn bộ quá trình mã hóa. Mặt khác, với sự bùng nổ của các nền tảng và thiết bị truy cập Internet, công nghệ Jamstack nổi lên như một giải pháp tối ưu cho việc xây dựng các ứng dụng web nhanh, bảo mật và dễ mở rộng.

Bài tiểu luận này được thực hiện với mục tiêu tìm hiểu tổng quan về kiến trúc Jamstack, cũng như ứng dụng công nghệ này để xây dựng một ứng dụng web minh họa trực quan quá trình khóa mở rộng trong thuật toán AES-128. Thông qua đó, người đọc không chỉ tiếp cận được với một mô hình phát triển web hiện đại mà còn hiểu sâu hơn về cách mà một thuật toán mã hóa phổ biến vận hành ở cấp độ thấp.

Nhóm chúng em hy vọng rằng rằng nội dung bài tiểu luận sẽ mang lại góc nhìn thực tiễn và dễ tiếp cận cho cả hai khía cạnh: kiến trúc web hiện đại và mã hóa dữ liệu – hai lĩnh vực đang ngày càng gắn bó chặt chẽ trong thế giới công nghệ ngày nay.

**MỤC LỤC**

# TỔNG QUAN

## Giới thiệu

Trong thời đại công nghệ thông tin phát triển mạnh mẽ, việc đảm bảo an toàn, bảo mật thông tin và kiến trúc phát triển web trở thành một trong những yêu cầu cấp thiết. Các hệ thống phần mềm, mạng máy tính, cũng như các dịch vụ trực tuyến đều đối mặt với nhiều mối đe dọa về bảo mật như đánh cắp dữ liệu, tấn công từ chối dịch vụ, hay xâm nhập trái phép.

Để đối phó với những nguy cơ đó, các cơ chế bảo mật ngày càng được chú trọng và cải tiến. Trong số đó, các kỹ thuật mã hóa dữ liệu đóng vai trò then chốt, đảm bảo dữ liệu được bảo vệ trong quá trình lưu trữ và truyền tải.

## Tổng quan về Jamstack

### Giới thiệu

JAMstack là một kiến trúc phát triển web hiện đại tập trung vào việc sử dụng JavaScript phía client, API có thể tái sử dụng và Markup đã được xây dựng sẵn. Khi đề cập đến khái niệm "Stack", chúng ta không còn nói về hệ điều hành, máy chủ web, ngôn ngữ backend hay cơ sở dữ liệu cụ thể. JAMstack không dựa vào công nghệ cụ thể nào mà thay vào đó, nó đại diện cho một phương thức mới để xây dựng các trang web và ứng dụng, mang lại hiệu suất cao hơn, bảo mật tốt hơn, chi phí mở rộng thấp hơn và một trải nghiệm phát triển tuyệt vời.

### Jamstack là gì?

Bạn có thể đã quen thuộc với các thuật ngữ như **MEAN stack**, **LAMP stack**, hay **MERN stack**, đây là những thuật ngữ dùng để nhóm các công nghệ cụ thể nhằm đạt được mục tiêu phát triển ứng dụng web. Tương tự như vậy, **JAMstack** là một kiến trúc phát triển web mới, được viết tắt từ ba yếu tố chính:

* **JavaScript**: Dùng để xử lý các tính năng động phía client (người dùng).
* **API**: Các API cung cấp dữ liệu hoặc thực hiện các tác vụ từ các dịch vụ bên ngoài.
* **Markup**: Là các tệp HTML đã được tạo sẵn trong quá trình triển khai, thay vì được sinh ra trong thời gian chạy.

A diagram of a software application

AI-generated content may be incorrect.

Các trang web truyền thống hoặc hệ thống quản lý nội dung (CMS) như WordPress, Drupal, v.v., thường phụ thuộc vào máy chủ, plugin và cơ sở dữ liệu để xử lý các yêu cầu động và tạo nội dung. Những hệ thống này phải thực hiện truy vấn dữ liệu trực tiếp từ cơ sở dữ liệu và tạo HTML trong thời gian chạy (runtime), điều này có thể ảnh hưởng đến hiệu suất và khả năng mở rộng.

Trong khi đó, JAMstack không yêu cầu các yếu tố này. Thay vào đó, nó sử dụng JavaScript để lấy dữ liệu từ API, phân phối các tệp tĩnh qua mạng phân phối nội dung (CDN) và tạo Markup (HTML) trong quá trình triển khai thông qua các công cụ tạo trang tĩnh. Phương pháp này giúp tăng tốc độ tải trang, cải thiện bảo mật bằng cách giảm các điểm tấn công từ cơ sở dữ liệu và máy chủ, đồng thời giảm chi phí vận hành nhờ vào việc sử dụng các tệp tĩnh và CDN để phân phối nội dung.

### Tại sao nên chọn Jamstack

* **Jamstack nhanh**  
   Khi nói đến việc giảm thời gian tải, không gì có thể so sánh với việc cung cấp các tệp có sẵn qua CDN. Các trang web JAMstack cực kỳ nhanh chóng vì HTML đã được tạo sẵn trong quá trình triển khai và chỉ cần được phân phối qua CDN mà không có bất kỳ sự chậm trễ nào.
* **Jamstack có tính bảo mật cao**  
   Mọi thứ trong JAMstack đều hoạt động thông qua API, do đó, không có sự phụ thuộc vào cơ sở dữ liệu hoặc các điểm yếu bảo mật truyền thống. Với việc các quy trình phía máy chủ được trừu tượng hóa thành các API dịch vụ vi mô, bề mặt tấn công được giảm thiểu, giúp trang web của bạn trở nên bảo mật hơn.
* **Jamstack chi phí thấp và dễ dàng mở rộng**  
   Các trang web JAMstack chỉ chứa một vài tệp với kích thước tối thiểu có thể được phục vụ ở bất kỳ đâu. Việc thu nhỏ và phục vụ các tệp này qua CDN hoặc các dịch vụ khác giúp việc mở rộng trở nên đơn giản và tiết kiệm chi phí.

### Quy trình làm việc của website tĩnh và website động

A screenshot of a computer

AI-generated content may be incorrect.

Đối với các website truyền thống sử dụng LAMP stack, lập trình viên viết mã và triển khai trên máy chủ. Khi người dùng gửi yêu cầu từ trình duyệt, yêu cầu đó được chuyển đến máy chủ để xử lý, sau đó máy chủ truy vấn cơ sở dữ liệu, tạo HTML trong thời gian thực và gửi kết quả về trình duyệt. Quy trình này phụ thuộc nhiều vào tốc độ phản hồi của máy chủ và hiệu năng của cơ sở dữ liệu.

Trong khi đó, đối với website tĩnh theo kiến trúc JAMstack, lập trình viên viết mã và nội dung được xây dựng sẵn rồi lưu trữ trên CDN. Khi người dùng truy cập website, các tệp HTML, CSS, JavaScript đã được dựng trước đó sẽ được tải về ngay lập tức từ CDN, giúp giảm thời gian phản hồi và tăng tốc độ truy cập. Kiến trúc nhỏ gọn và không phụ thuộc vào xử lý phía máy chủ cũng giúp JAMstack tối ưu hóa tốt cho SEO kỹ thuật và trải nghiệm người dùng.

* **JAMstack có thực sự là "tĩnh"?**

Mặc dù sản phẩm cuối cùng của một website JAMstack là các tệp tĩnh, nhưng nó vẫn có thể thực hiện các chức năng động như biểu mẫu liên hệ, phần nhận xét, thanh toán trực tuyến... Những chức năng này được xử lý thông qua các API của bên thứ ba hoặc các dịch vụ đám mây (serverless functions).

Nói cách khác, trong kiến trúc JAMstack, “tĩnh” không có nghĩa là cố định hay không thay đổi, mà là việc nội dung và cấu trúc trang được tạo sẵn một lần trong quá trình triển khai. Sau đó, các tương tác động sẽ được bổ sung bằng JavaScript và API, mang đến trải nghiệm linh hoạt nhưng vẫn giữ được lợi thế về hiệu suất và bảo mật. Đây có thể được xem là một thế hệ mới của website tĩnh – hiện đại, nhanh, và phù hợp với nhu cầu kinh doanh ngày nay.

### Các phương pháp tối ưu khi phát triển website với JAMstack

* Sử dụng CDN để phân phối tệp thay vì máy chủ truyền thống.
* Đơn giản hóa cài đặt và đóng góp bằng npm, Git để thiết lập nhanh chóng.
* Dùng công cụ build như Babel, Webpack để hỗ trợ đa trình duyệt.
* Đảm bảo tiêu chuẩn web và khả năng truy cập tốt cho mọi người dùng.
* Tự động hóa quy trình build để tiết kiệm thời gian và tránh lỗi.
* Triển khai tự động với nền tảng như Netlify để cập nhật nhanh chóng.

## Tổng quan về AES

### Giới thiệu

AES (Advanced Encryption Standard) là một thuật toán mã hóa hoạt động theo cơ chế mã hóa khối (block cipher). Đây là một trong những phương pháp mã hóa được sử dụng rộng rãi nhất hiện nay, sử dụng khóa đối xứng – nghĩa là cùng một khóa được dùng cho cả quá trình mã hóa và giải mã. Trong suốt quá trình hoạt động, AES sử dụng một khóa gốc (bí mật) để tạo ra một dãy các khóa con (round keys) thông qua quá trình gọi là mở rộng khóa (key expansion).

### Đặc điểm kĩ thuật

Một số khái niệm cơ bản:

* Bản rõ (Plaintext): Là thông tin gốc trước khi được mã hóa, có thể đọc và hiểu được.
* Bản mã (Ciphertext): Là dữ liệu đã được mã hóa từ bản rõ, dưới dạng không thể hiểu được nếu không có khóa giải mã.
* Khóa (Key): Là thông số bí mật được sử dụng trong quá trình mã hóa hoặc giải mã để bảo vệ dữ liệu.
* Mã hóa (Encryption): Là quá trình chuyển đổi bản rõ thành bản mã nhằm bảo vệ nội dung khỏi truy cập trái phép. Quá trình này có thể dùng hoặc không dùng khóa tùy theo phương pháp mã hóa.
* Giải mã (Decryption): Là quá trình khôi phục dữ liệu từ bản mã về dạng ban đầu là bản rõ.

AES (Advanced Encryption Standard) là một thuật toán mã hóa khối sử dụng cơ chế mã hóa đối xứng, nghĩa là cùng một khóa được dùng cho cả quá trình mã hóa và giải mã. Thuật toán này hỗ trợ ba độ dài khóa khác nhau: 128-bit, 192-bit và 256-bit, tương ứng với ba biến thể: AES-128, AES-192 và AES-256. Mỗi phiên bản sẽ thực hiện một số vòng lặp (round) khác nhau trong quá trình mã hóa:

* AES-128: 10 vòng
* AES-192: 12 vòng
* AES-256: 14 vòng

### Quy trình mã hóa

Quy trình mã hóa sử dụng bốn phép biến đổi chính:

* AddRoundKey: cộng ⊕ mã khóa của chu kỳ vào trạng thái hiện hành. Độ dài của mã khóa của chu kỳ bằng với kích thước của trạng thái.
* SubBytes: thay thế phi tuyến mỗi byte trong trạng thái hiện hành thông qua bảng thay thế (S-box).
* MixColumns: trộn thông tin của từng cột trong trạng thái hiện hành. Mỗi cột được xử lý độc lập.
* ShiftRows: dịch chuyển xoay vòng từng dòng của trạng thái hiện hành với di số khác nhau.

Quy trình mã hóa được tóm tắt lại như sau:

* Thực hiện thao tác AddRoundKey đầu tiên trước khi thực hiện các chu kỳ mã hóa.
* N−1N - 1N−1 chu kỳ mã hóa bình thường, mỗi chu kỳ bao gồm bốn bước biến đổi liên tiếp nhau: SubBytes, ShiftRows, MixColumns, và AddRoundKey.
* Thực hiện chu kỳ mã hóa cuối cùng: trong chu kỳ này thao tác MixColumns được bỏ qua.

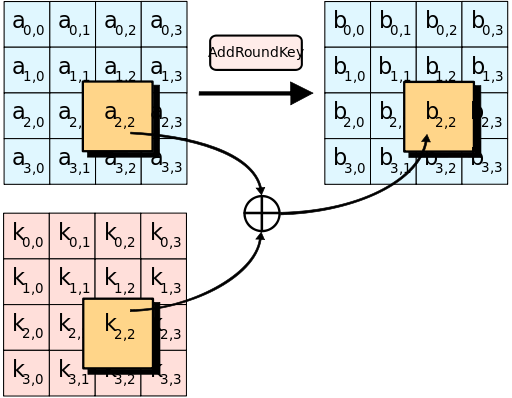
A diagram of a key register

AI-generated content may be incorrect.

Hình: Sơ đồ mã hóa AES

#### AddRoundKey

Khóa con được sinh ra từ khóa chính thông qua quá trình tạo khóa con Rijndael. Các khóa con này có cùng độ dài với khối dữ liệu và được sử dụng để kết hợp với khối trạng thái hiện tại. Việc kết hợp được thực hiện bằng cách thực hiện phép XOR giữa từng bit của khóa con và dữ liệu, đảm bảo rằng mỗi vòng mã hóa đều có ảnh hưởng của khóa, tăng tính bảo mật của thuật toán.



Hình: Thao tác của AddRoundkey

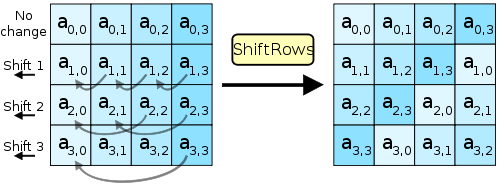
#### ShiftRows

Các hàng trong ma trận dữ liệu được dịch vòng một số bước nhất định.

* Hàng đầu tiên (hàng 0) không thay đổi, tức là các byte của hàng này được giữ nguyên vị trí.
* Hàng thứ hai (hàng 1) có các byte dịch vòng trái một vị trí, tức là byte đầu tiên sẽ được đưa ra sau cùng của hàng.
* Hàng thứ ba (hàng 2) được dịch vòng trái hai vị trí.
* Hàng thứ tư (hàng 3) sẽ được dịch vòng trái ba vị trí.

Như vậy, sau bước ShiftRows, mỗi cột trong khối dữ liệu đầu ra sẽ bao gồm các byte từ các hàng khác nhau của khối dữ liệu đầu vào.

Đối với Rijndael, vì có thể sử dụng các độ dài khối khác nhau, số lượng vị trí dịch vòng cũng sẽ thay đổi tương ứng. Trong các trường hợp này, số lượng dịch vòng của mỗi hàng sẽ phụ thuộc vào độ dài khối được sử dụng, điều này làm tăng sự linh hoạt của thuật toán Rijndael so với AES.



Hình: Thao tác của ShiftRows

#### MixColumns

Bước **MixColumns** thực hiện việc kết hợp bốn byte trong mỗi cột thông qua một phép biến đổi tuyến tính khả nghịch. Mỗi khối 4 byte đầu vào sẽ tạo ra một khối 4 byte đầu ra, trong đó mỗi byte đầu vào ảnh hưởng đến tất cả bốn byte đầu ra. Cùng với bước **ShiftRows**, bước **MixColumns** góp phần tạo ra tính chất khuếch tán cho thuật toán, giúp bảo vệ dữ liệu khỏi các tấn công phân tích.

Trong bước này, mỗi cột được xem như một đa thức trong trường hữu hạn, và được nhân với một đa thức đặc biệt:

c(x) = 3x3 + x2 + x + 2 mod (x + 1)

Điều này có nghĩa là phép biến đổi **MixColumns** có thể được xem như một phép nhân ma trận trong trường hữu hạn

A diagram of a complex number

AI-generated content may be incorrect.

Hình: Thao tác của MixColumns

#### SubBytes

Các byte trong quá trình mã hóa được thay thế thông qua một bảng tra S-box. Đây là một bước quan trọng của thuật toán, thực hiện phép biến đổi phi tuyến. S-box này được xây dựng từ một phép biến đổi nghịch đảo trong trường hữu hạn GF (28), mang tính chất phi tuyến. Để bảo vệ hệ thống khỏi các tấn công sử dụng đặc tính đại số, S-box được thiết kế bằng cách kết hợp phép nghịch đảo với một phép biến đổi affine khả nghịch. Đồng thời, S-box cũng được lựa chọn để tránh các điểm bất động (fixed points), giúp tăng tính bảo mật của thuật toán.

A diagram of a subbytes

AI-generated content may be incorrect.

Hình: Thao tác của SubBytes

A table of numbers and letters

AI-generated content may be incorrect.

Hình: Bảng S-Box

#### Thuật toán sinh Khóa (KeyExpansion)

Chức năng KeyExpansion thực hiện tính toán khóa vòng cho bước lặp mã hóa và bước tạo ngõ ra. Kết quả của một lần thực thi KeyExpansion là một khóa vòng sử dụng cho chức năng AddRoundKey. Với mã hóa AES-128, số khóa vòng là 10 tương ứng với 9 lần AddRoundKey ở bước lặp mã hóa và 1 lần AddRoundKey ở bước tạo ngõ ra.

Chức năng KeyExpansion được thực hiện thông qua 4 chức năng là RotWord, SubWord, AddRcon và AddW.

A diagram of a flowchart

AI-generated content may be incorrect.

Hình: Chức năng KeyExpansion

Quá trình tạo khóa:

Lấy cột cuối cùng trong ma trận khóa và dịch byte đầu xuống dưới cùng. Và thay thế các byte trong cột này bằng các byte trong S-Box nhu bước SubBytes, sau đó được XOR từng bit với cột đầu tiên và cột thứ i tương ứng với vòng mã hóa thứ i của ma trận Rcon. Kết quả là cột đầu tiên trong ma trận khóa kết quả của vòng i + 1.

A diagram of a block diagram

AI-generated content may be incorrect.

Sau đó ta XOR lần từng bit của các cột 2 -> 4 của ma trận khóa đầu vào, với cột vừa tạo để tạo ra các cột còn lại cho ma trận khóa vòng i + 1.

### Quy trình giải mã

Các bước tiến hành quá trình giải mã như sau:

* Bắt đầu bằng thao tác AddRoundKey, thực hiện trước khi vào các vòng giải mã chính.
* Tiếp theo là Nr - 1 vòng lặp giải mã tiêu chuẩn, mỗi vòng gồm bốn bước liên tiếp: InvShiftRows, InvSubBytes, AddRoundKey, và InvMixColumns.
* Trong vòng giải mã cuối cùng, chỉ thực hiện các bước như trên nhưng bỏ qua InvMixColumns.

#### InvShiftRows

Đây là thao tác đảo ngược của bước ShiftRows trong quá trình mã hóa. Các hàng trong ma trận dữ liệu sẽ được xoay vòng ngược lại theo hướng ngược với bước ShiftRows.

#### InvSubBytes

InvSubBytes là bước đảo của SubBytes. Thay vì dùng S-box, bước này sử dụng ma trận nghịch đảo của S-box (InvS-box) để thay thế các byte, nhằm khôi phục giá trị ban đầu trước mã hóa.

#### InvMixColumns

InvMixColumns thực hiện phép biến đổi nghịch đảo của MixColumns. Cột dữ liệu được nhân với một ma trận nghịch đảo đặc biệt trong trường hữu hạn, đảo ngược tác động của phép khuếch tán đã áp dụng trong bước MixColumns của quá trình mã hóa.

<https://vi.wikipedia.org/wiki/Advanced_Encryption_Standard>

https://topdev.vn/blog/gioi-thieu-ve-jamstack/