Thành viên nhóm :

Nguyễn Mạnh Cường

Phạm Văn Lộc

Phạm Văn Trung

Vũ Thành Trung

**BÁO CÁO2**

**MÃ HÓA FILE DỮ LIỆU LỚN**

**Mục lục**

**Chương 1: Tổng quan vấn đề...................................................1**

* 1. Bài toán................................................................................2
  2. Thế giới đã làm gì................................................................2
  3. Vấn đề chưa giải quyết........................................................3
  4. Đề xuất bản thân..................................................................5

**Chương 2: Nội dung thực hiện.................................................6**

2.1 Sử dụng HDFS.....................................................................6

2.2 Sử dụng RSA.......................................................................8

2.3 Sử dụng AES........................................................................9

**Chương 3: Kết quả thực hiện..................................................11**

3.1 Giải quyết vấn đề...............................................................11

3.2 Kết quả...............................................................................12

3.3 Kiểm thử............................................................................12

**Chương 4: Kết luận kiến nghị.................................................13**

4.1 Nhóm đã làm được gì........................................................13

4.2 Tồn đọng............................................................................13

4.3 Hướng phát triển, nâng cấp................................................14

\*Tài liệu tham khảo/Kết luận.....................................................14

**Chương 1: Tổng quan vấn đề**



1.1.Bài toán

-Dữ liệu lớn là một thuật ngữ cho việc xử lý một tập hợp dữ liệu rất lớn và phức tạp mà các ứng dụng xử lý dữ liệu truyền thống không xử lý được. Dữ liệu lớn bao gồm các thách thức như phân tích, thu thập, giám sát dữ liệu, tìm kiếm, chia sẻ, lưu trữ, truyền nhận, trực quan, truy vấn và tính riêng tư.

-Chính vì nó liên quan tới rất nhiều dữ liệu nên chúng ta mới cần phải bảo vệ cụ thể là mã hóa.

1.2.Thế giới đã làm gì

-Trên thế giới có vô vàn cách mã hóa dữ liệu lớn với sử dụng thư viện HDFS hoặc thuật toán AES.

-Mọi chi tiết khó khăn đã có thuật toán AES lo vì Thuật toán mã hoá khối mạng thay thế hoán đổi (SPN) được áp dụng vào trong chuẩn mã hoá AES, với mỗi khối kích thước là 128, 192 và 256 bit. Quá trình khởi đầu với khối [plain text](https://vi.wikipedia.org/wiki/V%C4%83n_b%E1%BA%A3n_thu%E1%BA%A7n_t%C3%BAy) (văn bản thuần túy) kích thước chuẩn được chèn vào hàng, và cuối cùng là mã hoá. Ba bước không thể thiếu đó là **thay thế**, **chuyển đổi** và**hòa trộn**.



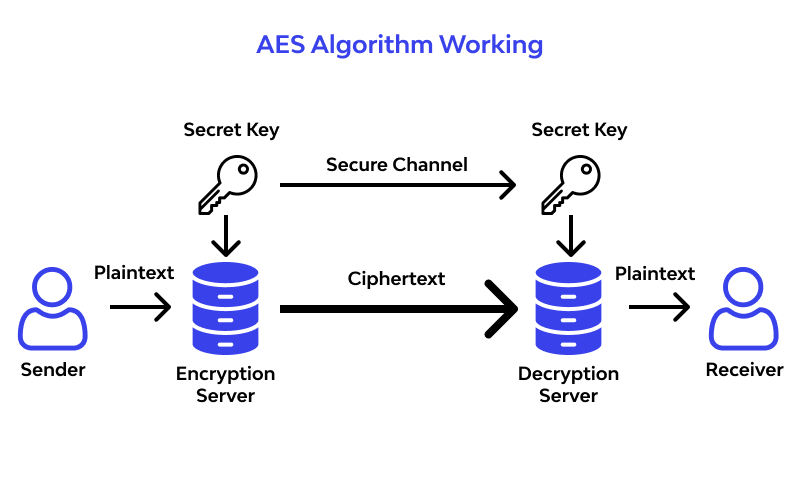
-Chính vì thuật toán này có thể mã hóa khối nên việc đọc ghi từng byte đơn sơ không cần tới thư viện HDFS.

1.3.Vấn đề chưa giải quyết tốt

-Mỗi giây, trên các CPU yếu hơn có thể là 10 giây một lần, bạn sẽ thấy một số có 9 chữ số. Trong 1 ngày bạn sẽ thấy các số có 10 chữ số. Trong 100 ngày, bạn sẽ thấy các số có 12 chữ số. Trong 3 năm, bạn sẽ thấy các số có 13 chữ số. Nếu bạn trả tiền cho 1 000 máy tính trong 30 năm, thì bạn sẽ nhận được các số có 16 chữ số. Vẫn còn rất xa những con số 77 chữ số.

-Kịch bản này rất đơn giản và nhanh chóng. Nhưng brute-forceing của AES sẽ chậm hơn nhiều. Đối với mọi ứng cử viên chính, người ta sẽ cần giải mã một số dữ liệu đáng kể, sau đó phân tích xem dữ liệu được giải mã có nghĩa là bất kỳ thứ gì như tài liệu, hình ảnh, video hay không.

-Hãy thực tế. Ước tính số tiền mà kẻ tấn công sẽ thu được nếu tệp của bạn bị cưỡng bức thô bạo và liệu chúng có thể bán tệp của bạn cho ai đó hay không. Họ sẽ nhận được 1 000 đô la , 1 000 000 đô la chứ? Không ai có thể brute-force mật khẩu AES của bạn (ngoại trừ việc kiểm tra từ điển với mật khẩu yếu điển hình và các biến thể của chúng, rất nhanh chóng). Nếu bí mật của bạn trị giá 1 000 000 đô la , thì việc trả cho tin tặc 500 000 đô la để đột nhập vào hệ thống của bạn, đọc mật khẩu của bạn hoặc cài đặt trình ghi khóa có thể rẻ hơn nhiều; hoặc cài đặt camera gián điệp trong nhà, ô tô của bạn, v.v. và do đó lấy được mật khẩu của bạn; hoặc đánh cắp chìa khóa của bạn nếu bạn có mã thông báo phần cứng.



1.4.Đề xuất của bản thân

-Có lẽ việc tôi có thể làm để khuyên bạn là hãy bảo mật được thông tin bằng mã hóa AES đó là hãy bảo vệ mật khẩu của mình, vì với trí thông minh của con người thì không gì là không thể.

-Chỉ cần một tập thể hacker làm việc thì mọi dữ liệu trên Internet sẽ bị đánh cắp, bạn nên nhớ mọi thứ trên Internet không bao giờ có thể bảo mật hoàn toàn vẫn sẽ có các lỗ hỏng để xâm nhập.

-Tốt nhất vẫn là không nên nói với ai về thứ mà bạn có được.

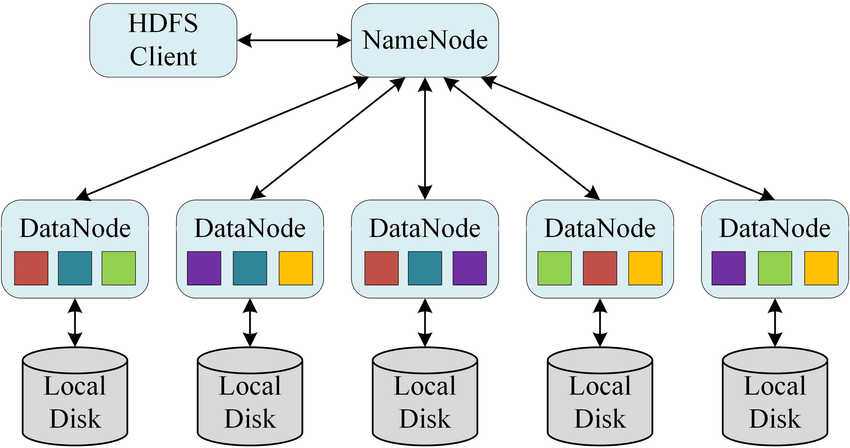


**Chương 2. Nội dung thực hiện**

**-**Nhóm đã lên các phương án đối với việc đọc ghi dữ liệu lớn tham khảo trên mạng.

**-Phương án 1 : Sử dụng HDFS**

-­HDFS là viết tắt của Hadoop Distributed File System và nó là một hệ thống lưu trữ chính được dùng bởi Hadoop. [Nó cung cấp truy cập hiệu suất cao đến dữ liệu trên các cụm Hadoop](https://itnavi.com.vn/blog/hdfs-la-gi/).



-Ưu điểm của HDFS:

* Chi phí thấp: các DataNodes lưu trữ dữ liệu trên các phần cứng với chi phí thấp .
* Bộ dữ liệu lưu trữ lớn: dữ liệu được lưu trữ trên HDFS có kích thước lớn từ megabyte đến petabyte và tồn tại bất kỳ định dạng nào, bao gồm cả dữ liệu có cấu trúc và phi cấu trúc .
* Cho phép dữ liệu có thể phân tán: Nếu như có một cụm Hadoop mà trong đó bao gồm nhiều máy tính thì bạn chỉ cần đưa một file dữ liệu vào HDFS. [Khi đó, thì file sẽ tự động được chia nhỏ thành nhiều phần rồi được lưu trữ ở các máy tính đó](https://itnavi.com.vn/blog/hdfs-la-gi/).
* [Cho phép tính toán và phân tán song song: Thay vì chỉ sử dụng một máy để xử lý công việc, thì với HDFS thì bạn có thể để các máy hoạt động song song để xử lý chung một công việc để tiết kiệm thời gian](https://itnavi.com.vn/blog/hdfs-la-gi/).
* [Cho phép nhân bản các file: Đặc điểm này sẽ giúp bạn đề phòng được các trường hợp một máy tính trong cụm Hadoop phát sinh sự cố thì dữ liệu sẽ được backup lại mà không bị mất](https://itnavi.com.vn/blog/hdfs-la-gi/).

-Nhược điểm của HDFS

* Cloud Storage có thể làm tăng độ lệch I/O: Trong nhiều tình huống, Cloud Storage có tốc độ I/O cao hơn HDFS. [Điều này có thể có vấn đề nếu bạn yêu cầu I/O nhất quán, chẳng hạn như ứng dụng hỗ trợ HBase hoặc cơ sở dữ liệu NoSQL khác](https://blog.cloud-ace.vn/hdfs-vs-cloud-storage-uu-khuyet-diem-va-meo-chuyen-doi/).
* Cloud Storage không hỗ trợ nối hoặc cắt ngắn nội dung trong file: Trên Cloud Storage, đối tượng là bất biến có nghĩa là các đối tượng được tải lên không được thay đổi trong suốt quá trình lưu trữ của nó. Trong thực tế, điều này có nghĩa là bạn có thể thực hiện các thay đổi nội dung của đối tượng như nối-cắt nội dung của đối tượng. [Bạn chỉ có thể ghi đè đối tượng](https://blog.cloud-ace.vn/hdfs-vs-cloud-storage-uu-khuyet-diem-va-meo-chuyen-doi/).
* Cloud Storage không tuân thủ POISIX: Công bằng mà nói thì HDFS cũng không hoàn toàn tuân thủ POISIX. [Tuy nhiên, qua nhiều năm, HDFS đã tích hợp được nhiều tính năng giống POISIX và các nhà phát triển đã xây dựng nên các ứng dụng Hadoop và Spark để tận dụng các tính năng có thể không hoạt động tốt trên Cloud Storage](https://blog.cloud-ace.vn/hdfs-vs-cloud-storage-uu-khuyet-diem-va-meo-chuyen-doi/).
* Cloud Storage không hiển thị tất cả thông tin hệ thống của đối tượng: Nếu bạn chạy lênh như “hadoop fsck -files -blocks” trong thư mục trên HDFS. Bạn sẽ nhìn thấy những thông tin hữu ích như về sức khỏe của các blocks. [Cloud Storage trừu tượng hóa toàn bộ lớp quản lý của tất cả lớp lưu trữ](https://blog.cloud-ace.vn/hdfs-vs-cloud-storage-uu-khuyet-diem-va-meo-chuyen-doi/).
* [Cloud Storage có thể có độ trễ lớn: Cloud Storage có thể có độ trễ cao hơn so với HDF](https://blog.cloud-ace.vn/hdfs-vs-cloud-storage-uu-khuyet-diem-va-meo-chuyen-doi/).

=>Nhóm đề ra phương án sử dụng thư viện HDFS để đọc ghi file sau đó thay đổi giá trị bit để mã hóa file.

-**Phương án 2 : Sử dụng RSA**

**-**RSA là một thuật toán mật mã hóa khóa công khai trong mật mã học. Đây là thuật toán đầu tiên phù hợp với việc tạo ra chữ ký điện tử đồng thời với việc mã hóa. Nó đánh dấu một sự tiến bộ vượt bậc của lĩnh vực mật mã học trong việc sử dụng khóa công cộng . RSA đang được sử dụng phổ biến trong thương mại điện tử và được cho là đảm bảo an toàn với điều kiện độ dài khóa đủ lớn .



-Ưu điểm của RSA

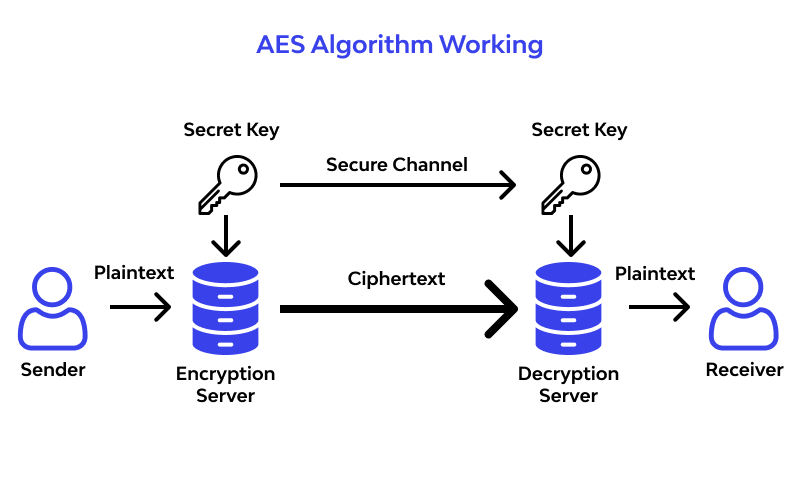
* Tốc độ mã hóa nhanh do kích thước khóa tương đối ngắn .
* Tốc độ xử lý nhanh .
* Dễ dàng trao đổi khóa .

-Nhược điểm của RSA

* Một nhược điểm lớn của thuật toán RSA đó là tốc độ mã hóa chậm hơn nhiều so với thuật toán mã hóa DES. [Do vậy trong một vài trường hợp để tăng tốc độ, người mã hoá sẽ mã hoá tài liệu bằng một thuật toán mã hóa khác nhưng khóa sẽ được mã hoá bằng RSA](https://viblo.asia/p/rsa-va-nhung-dieu-thu-vi-m68Z0apAlkG).

-**Phương án 3 : Sử dụng AES**

**-** AES (viết tắt của từ tiếng anh: Advanced Encryption Standard, hay Tiêu chuẩn mã hóa nâng cao) là một thuật toán mã hóa khối được chính phủ Hoa Kỳ áp dụng làm tiêu chuẩn mã hóa. Thuật toán được xây dựng dựa trên Rijndael Cipher phát triển bởi 2 nhà mật mã học người Bỉ: Joan Daemen và Vincent Rijmen .



-Ưu điểm của AES

* Tính bảo mật cao: AES là một thuật toán mã hóa đối xứng mạnh, có khả năng đảm bảo tính bảo mật cao trong việc mã hóa và giải mã dữ liệu .
* Tốc độ nhanh: Thuật toán này có tốc độ mã hóa và giải mã rất nhanh, phù hợp với các ứng dụng đòi hỏi tốc độ cao như truyền dữ liệu qua mạng .
* Linh hoạt: AES cũng có thể sử dụng các khóa với độ dài khác nhau (128, 192 hoặc 256 bit) để tăng cường tính bảo mật .

-Nhược điểm của AES

* Quá trình trao đổi khóa phải được thực hiện thông qua một hệ thống mã hóa khóa công khai khác .
* Cũng như một số thuật toán khác, AES cũng bị tác động với phương pháp tấn công kênh bên (side channel attack) tức là việc tấn công lên các hệ thống thực hiện thuật toán có nguy cơ làm lộ dữ liệu .
* Chính vì việc mô tả thuật toán khá đơn giản nên nhiều nhà mật mã học lo sợ rằng AES sẽ bị tấn công trong một tương lai không xa với sự giúp sức của các công nghệ hiện đại hơn .
* Chính vì thuật toán đơn giản của AES và tốc độ tối ưu nên tôi cũng không nghĩ bạn cần RSA. Bạn sẽ lưu trữ khóa cá nhân ở đâu và làm cách nào để bảo vệ nó khỏi bị đánh cắp? Sử dụng một mật khẩu tốt và khó đoán, cũng như một lượng muối (Trong mật mã, muối là dữ liệu ngẫu nhiên được sử dụng làm đầu vào bổ sung cho hàm một chiều có chức năng băm dữ liệu, mật khẩu hoặc cụm mật khẩu. Muối được sử dụng để bảo vệ mật khẩu trong lưu trữ )đủ dài và sử dụng một KDF tốt để lấy khóa AES nhằm mã hóa dữ liệu của bạn, tôi nghĩ nó sẽ hoạt động tốt hơn? Và 256 bit là rất lớn, sẽ mất nhiều thời gian hơn so với tuổi của vũ trụ hiện tại, sử dụng mọi máy tính trên thế giới này để bẻ khóa thành công AES 256 bit.

**Chương 3. Kết quả thực hiện**

3.1.Cách giải quyết vấn đề

**-**Khi làm việc với một large file,big data muôn vàn vấn đề xảy đến trong quá trình tìm hiểu cũng như thực thi. Nào là đọc ghi file như nào, thuật toán ra sao khi chưa được học ít nào với chương trình dạy học trên trường.Sự mù mờ đến từ việc nhảy cóc chương trình sinh viên đã phải đối mặt với Cấu trúc dữ liệu và giải thuật trong khi lập trình hướng đối tượng còn chưa được học.Nhưng từ đó mọi sự khó khăn sẽ được giải quyết khi bạn chịu khó tìm hiểu. Trên mạng không hề thiếu những source code với mọi thuật toán nhưng không chỉ dựa vào copy paste nhóm mang code về còn phải chỉnh sửa sao cho hợp lý, thích hợp môi trường.

Lẫn đến việc xây dựng giao diện Winform , 1 chàng trai mới bước chân viết ứng dụng đến việc code để lắng nghe button còn mới mẻ chứ chưa nói gì đến việc chỉnh sửa 1 thao tác nút bấm nào đó là đã không chạy được form.

3.2.Kết quả

-Không bõ công ngày đêm tìm tòi thì cuối cùng nhóm cũng hoàn thành bài tập, khó ở đây không phải làm thuật toán khó mà khó đó chính là bạn đưa cho người ta một cục đá và bảo người ta chặt cái cây. Sự tìm hiểu vỡ lẽ giúp các thành viên phần nào có được kinh nghiệm cho bản thân, trau dồi kiến thức với ngành nghề. Chỉ sau một lần chạy code mọi thứ đều diễn ra như những gì mà nhóm mong đợi. Từ việc nhập file đến nhập mật khẩu, mã hóa và giải mã.

3.3.Kiểm thử

-Như mọi thứ tôi Nhóm đã đề cập ở trên, giải đoạn kiểm thử mọi thứ đều ổn với file 15MB-100MB. Với file lớn hơn 1GB nhóm chưa thử tới mong rằng nó vẫn ổn với thuật toán AES này.

**Chương 4. Kết luận kiến nghị**

4.1.Nhóm đã làm được gì

-Có lẽ điều đơn giản nhất mà nhóm làm được đó là mã hóa được dữ liệu trung bình, khá. Còn với dung lượng lớn đến mức hơn 1GB có lẽ nhóm sẽ thử sau nhưng mục đích trước mắt của nhóm là như vậy.

4.2.Tồn đọng

-Về an ninh của AES thì các nhà khoa học đánh giá là chưa cao. Họ cho rằng ranh giới giữa số chu trình của thuật toán và số chu trình bị phá vỡ quá nhỏ. Nếu các kỹ thuật tấn công được cải thiện thì AES có thể bị phá vỡ. Ở đây, phá vỡ có nghĩa chỉ bất cứ phương pháp tấn công nào nhanh hơn tấn công kiều duyệt toàn bộ. Vì thế một tấn công cần thực hiện 2120 cũng được coi là thành công mặc dù tấn công này chưa thể thực hiện trong thực tế. Tại thời điểm hiện nay, nguy cơ này không thực sự nguy hiểm và có thể bỏ qua. Tấn công kiểu duyệt toàn bộ quy mô nhất đã từng thực hiện là do distributed.net thực hiện lên hệ thống 64 bít RC5 vào năm 2002 (Theo định luật Moore thì nó tương đương với việc tấn công vào hệ thống 66 bít hiện nay).

-Một vấn đề khác nữa là cấu trúc toán học của AES. Không giống với các thuật toán mã hóa khác, AES có mô tả toán học khá đơn giản. Tuy điều này chưa dẫn đến mỗi nguy hiểm nào nhưng một số nhà nghiên cứu sợ rằng sẽ có người lợi dụng được cấu trúc này trong tương lai.

-Một ví dụ điển hình là Vào năm 2002, Nicolas Courtois và Josef Pieprzyk phát hiện một tấn công trên lý thuyết gọi là tấn công XSL và chỉ ra điểm yếu tiềm tàng của AES. Tuy nhiên, một vài chuyên gia về mật mã học khác cũng chỉ ra một số vấn đề trong cơ sở toán học của tấn công này và cho rằng các tác giả đã có sai lầm trong tính toán.

4.3.Phương hướng triển khai,nâng cấp.

-Nhóm không có thêm sự nâng cấp nào trong thuật toán này ngoài việc sử dụng một mã muối dài.

\*Tài liệu tham khảo: wikipedia,stackoverflow,codelearn,bing ai,CrytoAES.

**Kết luận**

Không thể phủ nhận là thuật toán DES, Triple DES, AES có nhiều ứng dụng trong viễn thông và công nghệ thông tin, việc làm chủ và cứng hóa các thuật toán rất có ý nghĩa đối với sự an toàn trong các giao dịch trên mạng, nhất là trong ứng dụng TMĐT. Nhìn chung, đối với Việt Nam, việc làm chủ công nghệ cứng hóa thuật toán DES, Triple DES, AES rất có ý nghĩa trong việc đảm bảo an toàn giao dịch trên mạng, đảm bảo an toàn trong truyền tin cho các đơn vị cơ yếu tại Việt Nam. Hiện nay, nhiều chuyên gia đang triển khai cứng hóa thuật toán DES, Triple DES, AES nhờ các công nghệ thiết kế số hiện đại, các chương trình và mạch phần cứng. Trên đây là bài thảo luận của nhóm chúng tôi. Trong quá trình tìm hiểu tài liệu và sự hiểu biết của nhóm về đề tài còn ít nên không thể không tránh khỏi những sai sót.