

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG**



**BÁO CÁO ĐỒ ÁN
MẬT MÃ HỌC**

**ĐỀ TÀI: BẢO MẬT VÀ KIỂM SOÁT QUYỀN TRUY CẬP
DỮ LIỆU TRÊN AMAZON RDS MYSQL**

Confidentiality and Access Control in Amazon RDS MySQL

Giảng viên hướng dẫn: TS. NGUYỄN NGỌC TỰ

Thực hiện bởi Nhóm 2, gồm:

- | | | |
|-------------------|----------|-------------|
| 1. LẠI QUAN THIỀN | 22521385 | Trưởng nhóm |
| 2. ĐẶNG ĐỨC TÀI | 22521270 | Thành viên |
| 3. LÊ MINH QUÂN | 22521181 | Thành viên |

Lớp: NT219.O21.ANTT

TP. Hồ Chí Minh, tháng 06 năm 2024

LỜI MỞ ĐẦU

Trong thời đại số hóa hiện nay, dữ liệu của các công ty, doanh nghiệp đều tăng liên tục theo cấp số nhân. Chính vì thế, nhu cầu lưu trữ dữ liệu ngày càng tăng cao. Thế nhưng, việc chạy đua phần cứng đòi hỏi chi phí cao đối với các doanh nghiệp nói chung và các doanh nghiệp vừa và nhỏ nói riêng. Cloud computing ra đời đã thay đổi cách chúng ta lưu trữ và chia sẻ dữ liệu, mang đến giải pháp lưu trữ chi phí thấp và dễ dàng truy cập thông qua các dịch vụ như Dropbox và Google Drive. Thế nhưng, điều này lại đặt là một thách thức lớn về việc bảo vệ thông tin, dữ liệu. Việc lưu trữ và truy xuất các thông tin trong cơ sở dữ liệu đòi hỏi phải có các cơ chế bảo mật mạnh mẽ nhằm đảm bảo tính bảo mật và toàn vẹn của dữ liệu.

Những tai nạn, sự cố về việc lộ thông tin cá nhân và dữ liệu đã giấy lên hồi chuông cảnh báo về tầm quan trọng trong việc bảo mật thông tin. Trong lĩnh vực y tế, dữ liệu đóng vai trò quan trọng trong việc quản lý hồ sơ bệnh nhân và vận hành các hoạt động của bệnh viện. Vì vậy, việc bảo vệ cơ sở dữ liệu trở thành một trong những ưu tiên hàng đầu của các tổ chức y tế.

Để giải quyết những lo ngại này, việc triển khai các cơ chế bảo mật nhằm cung cấp mức độ bảo vệ và kiểm soát truy cập cao hơn là rất cần thiết. Chính vì thế, nhóm chúng em đã quyết định chọn đề tài "**BẢO MẬT VÀ KIỂM SOÁT QUYỀN TRUY CẬP TRÊN AMAZON RDS MYSQL**".

Hệ thống bảo mật cơ sở dữ liệu này sẽ được xây dựng trên nền tảng kết hợp giữa các kỹ thuật mã hóa AES, CP-ABE và cơ chế kiểm soát truy cập ABAC. Sự kết hợp này không chỉ nâng cao tính bảo mật của dữ liệu mà còn cho phép quản lý truy cập một cách chính xác và hiệu quả hơn.

Với hệ thống bảo mật tích hợp này, bệnh viện sẽ có thể đảm bảo rằng dữ liệu bệnh nhân luôn được bảo vệ một cách tối ưu, đồng thời quản lý truy cập của người dùng một cách linh hoạt và hiệu quả. Chúng em hy vọng rằng đề tài này sẽ đóng góp tích cực vào việc nâng cao tính bảo mật của các cơ sở dữ liệu y tế, giúp các bệnh viện tăng cường bảo vệ thông tin cá nhân và dữ liệu quan trọng của bệnh nhân.

LỜI CẢM ƠN

Lời đầu tiên, cho phép tập thể nhóm 2 đến từ lớp NT219.O21.ANTT xin gửi lời cảm ơn và tri ân sâu sắc đến thầy Nguyễn Ngọc Tự - Giảng viên môn Mật mã học, Khoa Mạng máy tính và Truyền Thông, Trường Đại học Công nghệ Thông tin, ĐHQG-HCM. Chúng em cảm ơn thầy vì sự tận tâm giúp đỡ, chỉ bảo và hướng dẫn trong quá trình thực hiện đồ án. Những buổi thảo luận và sự hướng dẫn của thầy đóng vai trò quan trọng trong việc hoàn thiện đồ án này. Chúng em chân thành cảm ơn thầy vì kiến thức và kinh nghiệm mà thầy đã chia sẻ với chúng em.

Chúng em cũng muốn gửi lời cảm ơn đến tất cả các thành viên trong nhóm đồ án. Sự đóng góp và nỗ lực của mỗi người trong việc tìm kiếm tài liệu, đưa ra ý tưởng và hoàn thiện đề tài đã tạo nên thành công của cuộc. Mặc dù kiến thức của chúng em còn hạn chế và không tránh khỏi những sai sót, nhưng sự đóng góp ý kiến của mọi người đã giúp chúng em hoàn thiện và cải thiện đề tài một cách tốt nhất.

Cuối cùng, vì thời gian và năng lực có hạn nên không thể tránh khỏi sai sót trong khi thực hiện đồ án học tập của chúng em. Rất mong sự góp ý và bổ sung của thầy để đề tài chúng em trở nên hoàn thiện hơn.

Một lần nữa, tập thể nhóm xin chân thành cảm ơn thầy!

TP. Hồ Chí Minh, tháng 06 năm 2024

Nhóm 2, lớp NT219.O21.ANTT

MỤC LỤC

LỜI MỞ ĐẦU	
CHƯƠNG 1: GIỚI THIỆU	1
1.1. Tổng quan đề tài	1
1.2. Đặt vấn đề.....	1
1.2.1. Kịch bản.....	1
1.2.2. Các bên liên quan.....	1
1.3. Các rủi ro bảo mật.....	2
1.3.1. Đối với tấn công đường truyền	2
1.3.2. Đối với giả mạo các bên liên quan	2
1.3.3. Đối với rủi ro lộ thông tin nội bộ	3
1.4. Mục tiêu bảo mật	3
1.4.1. Tính bảo mật	3
1.4.2. Tính toàn vẹn	3
1.4.3. Ủy quyền.....	3
CHƯƠNG 2: GIẢI PHÁP	4
2.1. AES-GCM (Advanced Encryption Standard - Galois/Counter Mode):.....	4
2.1.1. Tổng quan	4
2.1.2. Minh họa thuật toán	4
2.2. CP-ABE (Ciphertext-Policy Attribute-Based Encryption):	8
2.2.1. Tổng quan	8
2.2.2. Về FAME: Fast Attribute-based Message Encryption.....	9
2.2.3. Minh họa thuật toán	10
2.3. ABAC (Attribute-Based Access Control)	13
2.3.1. Xác định các thuộc tính	13
2.3.2. Xác định các chính sách	13

2.3.3. Thi hành chính sách	13
CHƯƠNG 3. TÀI NGUYÊN CỦA MÔ HÌNH	15
3.1. Công cụ và tài nguyên	15
3.1.1. Ngôn ngữ lập trình Python 3.10.11.....	15
3.1.2. Môi trường triển khai.....	15
3.1.3. Thư viện hỗ trợ	15
3.1.4. Cơ sở dữ liệu.....	18
3.2. Kiến trúc hệ thống	19
3.2.1. Tổng quan kiến trúc hệ thống	19
3.2.2. CA (Center of Authority):	19
CHƯƠNG 4. TRIỂN KHAI.....	22
4.1. Kịch bản triển khai.....	22
4.1.1. Data Owners	22
4.1.2. Data Users.....	22
4.1.3. CA (Center of Authority):	23
4.2. Vấn đề bảo mật và giải pháp của mô hình:.....	23
4.3. Kiểm thử và kết quả kiểm thử:	23
4.3.1. Data Owners	23
4.3.2. Data Users.....	26
4.3.3. CA (Center of Authority)	29
CHƯƠNG 5. TỔNG KẾT.....	31
5.1. Kết quả đạt được	31
5.2. Hạn chế	32
5.3. Lời kết.....	32
DANH MỤC TÀI LIỆU THAM KHẢO	33

MỤC LỤC HÌNH ẢNH

Hình 1: Tạo Key và IV bằng AES-GCM-256	4
Hình 2: Hàm Mã hóa AES-GCM-256	5
Hình 3: Mã hóa dữ liệu theo cột bằng AES-GCM-256	5
Hình 4: Hàm giải mã AES-GCM-256	6
Hình 5: Giải mã dữ liệu theo cột bằng AES-GCM-256	7
Hình 6: Tạo Master Key và Public Key	10
Hình 7: Tạo Secret Key	10
Hình 8: Tạo bảng để lưu Keys.....	11
Hình 9: Giải mã CP-ABE	12
Hình 10: Một policy được thiết lập sẵn cho ABAC	14
Hình 11: Tổng quan kiến trúc hệ thống.....	19
Hình 12: Triển khai mã hóa dữ liệu.....	23
Hình 13: Dữ liệu sau khi được mã hóa.....	24
Hình 14: Mã hóa Keys AES bằng CP-ABE	24
Hình 15: Các Keys AES sau khi được mã hóa	25
Hình 16: Đưa dữ liệu đã mã hóa lên Cloud.....	25
Hình 17: Đưa các Keys AES đã mã hóa lên Cloud.....	26
Hình 18: Yêu cầu Keys theo thuộc tính.....	26
Hình 19: Thực hiện giải mã các Keys nhận được	27
Hình 20: Giải mã dữ liệu thông qua AES-GCM-256	27
Hình 21: Dữ liệu sau khi được giải mã	28
Hình 22: Chứng chỉ cho Server (localhost).....	29
Hình 23: Chứng chỉ cho Server	30
Hình 24: Gửi và nhận Public Key, Secret Key	30

CHƯƠNG 1: GIỚI THIỆU

1.1. Tổng quan đề tài

- Tên đề tài: Bảo mật và Kiểm soát Quyền truy cập Dữ liệu trên Amazon RDS MySQL.
- Thời gian thực hiện: từ tháng 03/2024 đến tháng 06/2024.
- Link github: [Click here](#)

1.2. Đặt vấn đề

Ngày nay, việc sử dụng các dịch vụ đám mây để lưu trữ dữ liệu nhạy cảm và bảo mật ngày càng trở nên phổ biến. Tuy nhiên, những lợi ích về sự tiện lợi và khả năng mở rộng đi kèm với những rủi ro bảo mật đáng kể cần được giải quyết. Ví dụ, một bệnh viện cần lưu trữ thông tin bệnh nhân, hồ sơ khám bệnh, đội ngũ bác sĩ, nhân viên và các thông tin nhạy cảm khác trên đám mây sẽ dễ bị đe dọa bởi các mối nguy hiểm bảo mật như vi phạm dữ liệu và các mối đe dọa từ nội bộ.

1.2.1. Kịch bản

Trong kịch bản này, một bệnh viện đang sử dụng dịch vụ đám mây để lưu trữ các tài sản kỹ thuật số riêng tư của họ (hồ sơ bệnh nhân, tài liệu nội bộ, v.v.). Do đó, họ sẽ phải đối mặt với nhiều vấn đề bảo mật như vi phạm dữ liệu và các mối đe dọa từ bên trong.

1.2.2. Các bên liên quan

1.2.2.1. CA (*Center of Authority*)

Đây là bên đáng tin cậy trong hệ thống, chịu trách nhiệm cung cấp các khóa và chứng chỉ cho các bên khác. CA cung cấp public key cho Data-owners để mã hóa dữ liệu và tạo secret key cho Data-users để giải mã dữ liệu trong hệ thống.

1.2.2.2. Data Owners

Chủ sở hữu dữ liệu, là người thực hiện mã hóa dữ liệu trong cơ sở dữ liệu bằng AES trước khi đưa lên cloud. Sau đó, Data-owners sử dụng public key của CA để mã hóa key AES, key sau khi mã hóa cũng được lưu trên cloud cùng với cơ sở dữ liệu đã mã hóa.

1.2.2.3. Data Users

Người dùng dữ liệu trong cơ sở dữ liệu. Vì dữ liệu được lưu trữ trên cloud đã được mã hóa, Data-users muốn truy cập và giải mã dữ liệu cần phải có secret key, key này được tạo bởi CA khi Data-users yêu cầu và cung cấp thuộc tính cho CA.

1.2.2.4. Cloud

Đây là nơi lưu trữ cơ sở dữ liệu và khóa AES sau khi đã được mã hóa, đồng thời cung cấp khả năng truy xuất dữ liệu cho người dùng.

1.2.2.5. Threats

Tập hợp các mối đe dọa cho hệ thống – là bên tìm cách gây hại, tấn công vào hệ thống nhằm mục đích đánh cắp, phá hoại dữ liệu.

1.3. Các rủi ro bảo mật

Các rủi ro bảo mật đối với hệ thống có thể chia thành: bên ngoài (gồm có tấn công đường truyền, giả mạo các bên liên quan), bên trong (lộ thông tin nội bộ).

1.3.1. Đối với tấn công đường truyền

Giải pháp là thiết lập kết nối bảo mật bằng TLS/SSL trong mạng nội bộ.

1.3.2. Đối với giả mạo các bên liên quan

1.3.2.1. Giả mạo Center of Authority

Để cấp Public Key cho Data Owners, nếu Data Owners sử dụng key đó để mã hóa dữ liệu thì kẻ giả mạo có thể giải mã những dữ liệu đó bằng cách tự tạo Secret Key cho bản thân: giải pháp là tạo Certificate để xác thực CA, Certificate của CA được ký bởi chính nó.

1.3.2.2. Giả mạo Data Owners

Kẻ giả mạo có thể đưa dữ liệu rác, không chính xác, làm tổn bộ nhớ lên Cloud. Do đó CA sẽ xác thực Data Owners trước khi cho phép vào hệ thống (qua ABAC), tạo cơ chế để chỉ những Data Owners mới được đưa dữ liệu lên Cloud.

1.3.2.3. Giả mạo Data-users

Kẻ giả mạo có thể gửi những thuộc tính giả cho server để tạo Secret Key, từ đó lấy được dữ liệu... Do đó, phía CA sẽ xác thực Data Users trước khi tạo Secret Key (qua ABAC), nếu dữ liệu của User đó có trong hệ thống thì sẽ được chấp nhận truy cập vào.

1.3.3. Đối với rủi ro lộ thông tin nội bộ

- Cloud được coi là một đối tượng semi-trusted (tức là có sự tin tưởng đến một mức độ nhất định), mặc dù có thể coi là một đối tượng đáng tin cậy nhưng vẫn có xác xuất xảy ra những việc không mong muốn đến dữ liệu như làm rò rỉ thông tin hay sử dụng dữ liệu cho mục đích khác: giải pháp là mã hóa những dữ liệu quan trọng không muốn bên cloud biết, khi đó dù dữ liệu bị rò rỉ thì thiệt hại ở mức nhỏ vì những dữ liệu đó đã mã hóa không thể đọc được.

- Ngoài ra, nhóm chúng em cũng sử dụng dịch vụ Cloud do bên thứ 3 có uy tín cung cấp là Amazon AWS để đảm bảo được độ tin cậy cao hơn.

1.4. Mục tiêu bảo mật

1.4.1. Tính bảo mật

Đảm bảo rằng dữ liệu nhạy cảm được lưu trữ trên đám mây được bảo vệ khỏi truy cập hoặc tiết lộ trái phép.

1.4.2. Tính toàn vẹn

Đảm bảo rằng dữ liệu lưu trữ trên đám mây không bị can thiệp hoặc sửa đổi một cách trái phép.

1.4.3. Ủy quyền

Đảm bảo rằng người dùng truy cập dữ liệu trên đám mây có quyền truy cập và đặc quyền phù hợp.

CHƯƠNG 2: GIẢI PHÁP

2.1. AES-GCM (Advanced Encryption Standard - Galois/Counter Mode):

2.1.1. Tổng quan

AES-GCM là một thuật toán mã hóa đối xứng được thiết kế để cung cấp cả tính bảo mật và tính toàn vẹn của dữ liệu. Chế độ mã hóa này cung cấp xác thực và bảo mật, đảm bảo rằng dữ liệu không bị giả mạo trong quá trình truyền tải hoặc lưu trữ. Nhóm chúng em sử dụng mode GCM, với kích thước khóa là 256 bit.

- **Tính bảo mật:** AES-GCM sử dụng thuật toán mã hóa AES để bảo vệ dữ liệu khỏi việc truy cập trái phép.

- **Tính toàn vẹn và xác thực:** Sử dụng thuật toán Galois Message Authentication Code (GMAC) để đảm bảo rằng dữ liệu không bị thay đổi trong quá trình truyền tải.

- **Tốc độ:** AES-GCM hoạt động rất nhanh và hiệu quả, phù hợp cho các ứng dụng yêu cầu tốc độ cao như truyền thông mạng và lưu trữ dữ liệu.

2.1.2. Minh họa thuật toán

2.1.2.1. Thiết lập

- Bao gồm 2 bước:
 - + **Tạo khóa AES:** Key 256 bit.
 - + **Tạo nonces (số ngẫu nhiên):** IV 96 bit.
- Sau đó gộp cả 2 vào một chuỗi duy nhất và lưu chúng dưới dạng Base64 vào tệp được cung cấp.

```
def gen_key(self, columns, keyfile):  
    key_iv_pairs = []  
  
    for column in columns:  
        iv = os.urandom(12) # Tạo IV ngẫu nhiên 12 bytes  
        key = os.urandom(32) # Tạo key ngẫu nhiên 32 bytes  
        key_iv_pairs[column] = base.ByteToBase64(key + iv)  
  
    base.save_keys_iv_to_file(key_iv_pairs, keyfile)  
  
    print("Tạo key và IV thành công cho các cột!")
```

Hình 1: Tạo Key và IV bằng AES-GCM-256

2.1.2.2. Mã hóa

```
# Hàm mã hóa sử dụng AES GCM
def encrypt(self, plaintext, key_iv):
    key_iv_bytes = base.Base64ToByte(key_iv)
    key = key_iv_bytes[:32]
    iv = key_iv_bytes[32:44]
    aesgcm = AESGCM(key)
    plaintext_bytes = plaintext.encode('utf-8')
    ciphertext_with_tag = aesgcm.encrypt(iv, plaintext_bytes, None)
    ciphertext = base.ByteToBase64(ciphertext_with_tag)
    return ciphertext
```

Hình 2: Hàm Mã hóa AES-GCM-256

```
def encrypt_data(self, encrypted_csv_path, selected_columns, cols, plaintext_file, key_iv_file):
    key_iv_dict = base.read_key_iv_from_file(key_iv_file)

    # If 'all' is specified in choices, replace it with all columns
    if 'all' in selected_columns:
        selected_columns = cols

    try:
        with open(plaintext_file, 'r', newline='') as csvfile:
            reader = csv.DictReader(csvfile)
            with open(encrypted_csv_path, 'w', newline='') as output_file:
                writer = csv.DictWriter(output_file, fieldnames=reader.fieldnames)
                writer.writeheader()

                for row in reader:
                    encrypted_row = row.copy()
                    for choice in selected_columns:
                        if choice in row:
                            plaintext = row[choice]
                            if choice in key_iv_dict:
                                try:
                                    ciphertext = self.encrypt(plaintext, key_iv_dict[choice])
                                    encrypted_row[choice] = ciphertext
                                except Exception as e:
                                    print(f"Mã hóa thất bại cho cột {choice}: {e}")
                            else:
                                print(f"Không tìm thấy key và IV cho cột {choice}")
                    writer.writerow(encrypted_row)

    except Exception as e:
        print(f"Đã xảy ra lỗi: {e}")

    print(f"Mã hóa thành công! Dữ liệu mã hóa được lưu tại '{encrypted_csv_path}'!")
```

Hình 3: Mã hóa dữ liệu theo cột bằng AES-GCM-256

- read_key_iv_from_file(...) : đọc tệp chứa các cặp khóa và IV và trả về một dictionary.

- Mở tệp CSV nguồn để đọc dữ liệu và tệp CSV đích để ghi dữ liệu mã hóa.
 - csv.DictReader và csv.DictWriter được sử dụng để đọc và ghi dữ liệu theo định dạng dictionary.
 - Duyệt qua từng hàng (row) trong tệp CSV nguồn.
 - Với mỗi hàng, sao chép dữ liệu sang encrypted_row để giữ nguyên các cột không mã hóa.
 - Với mỗi cột trong selected_columns, kiểm tra xem cột đó có tồn tại trong hàng không. Nếu có:
 - + Lấy giá trị văn bản rõ (plaintext) theo từng cột.
 - + Kiểm tra xem có khóa và IV cho cột đó trong key_iv_dict không. Nếu có:
 - o Gọi hàm encrypt để mã hóa văn bản rõ, cập nhật giá trị mã hóa trong encrypted_row.
 - o Nếu xảy ra lỗi trong quá trình mã hóa, in thông báo lỗi.
 - + Nếu không có khóa và IV cho cột đó, in thông báo lỗi.
- => Mã hóa các cột được chọn trong một tệp CSV và lưu kết quả vào một tệp CSV mới, sử dụng khóa và IV từ tệp đã cung cấp trước đó.

2.1.2.3. Giải mã

```
# Hàm giải mã sử dụng AES GCM
def decrypt(self, ciphertext, key_iv):
    key_iv_bytes = base.Base64ToByte(key_iv)
    key = key_iv_bytes[:32]
    iv = key_iv_bytes[32:44]
    ciphertext = base.Base64ToByte(ciphertext)
    aesgcm = AESGCM(key)
    recovertext_bytes = aesgcm.decrypt(iv, ciphertext, None)
    recovertext = recovertext_bytes.decode('utf-8')
    return recovertext
```

Hình 4: Hàm giải mã AES-GCM-256

```

def decrypt_data(self, decrypted_csv_path, selected_columns, cols, encrypted_csv_path, keys_file):

    key_iv_dict = base.read_key_iv_from_file(keys_file)

    # If 'all' is specified in choices, replace it with all columns
    if 'all' in selected_columns:
        selected_columns = cols

    try:
        with open(encrypted_csv_path, 'r', newline='') as csvfile:
            reader = csv.DictReader(csvfile)
            with open(decrypted_csv_path, 'w', newline='') as output_file:
                writer = csv.DictWriter(output_file, fieldnames=reader.fieldnames)
                writer.writeheader()

                for row in reader:
                    decrypted_row = row.copy()
                    for choice in selected_columns:
                        if choice in row:
                            ciphertext_base64 = row[choice]
                            if choice in key_iv_dict:
                                try:
                                    recovertext = self.decrypt(ciphertext_base64, key_iv_dict[choice])
                                    decrypted_row[choice] = recovertext
                                except Exception as e:
                                    print(f"Giải mã thất bại cho cột {choice}: {e}")
                            else:
                                return
                    writer.writerow(decrypted_row)

    except ValueError as e:
        return

    print(f"Giải mã thành công! Dữ liệu giải mã được lưu tại '{decrypted_csv_path}'!\n")

```

Hình 5: Giải mã dữ liệu theo cột bằng AES-GCM-256

- Hàm **read_key_iv_from_file(...)** đọc tệp chứa các cặp khóa và IV và trả về một dictionary.
 - Mở tệp CSV đã mã hóa để đọc dữ liệu và tệp CSV đích để ghi dữ liệu giải mã.
 - csv.DictReader và csv.DictWriter được sử dụng để đọc và ghi dữ liệu theo định dạng dictionary.
 - Duyệt qua từng hàng (row) trong tệp CSV đã mã hóa.
 - Với mỗi hàng, sao chép dữ liệu sang decrypted_row để giữ nguyên các cột không giải mã.
 - Với mỗi cột trong selected_columns, kiểm tra xem cột đó có tồn tại trong hàng không. Nếu có:
 - + Lấy giá trị văn bản mã hóa (ciphertext_base64) từ cột.
 - + Kiểm tra xem có khóa và IV cho cột đó trong key_iv_dict không. Nếu có:
 - o Gọi hàm decrypt để giải mã văn bản mã hóa, cập nhật giá trị giải mã trong decrypted_row.
 - o Nếu xảy ra lỗi trong quá trình giải mã, in thông báo lỗi.

+ Nếu không có khóa và IV cho cột đó, dừng hàm và không thực hiện giải mã.
=> Giải mã các cột được chọn trong một tệp CSV đã mã hóa và lưu trữ dữ liệu giải mã vào một tệp CSV mới, sử dụng khóa và IV từ tệp đã cung cấp trước đó.

2.2. CP-ABE (Ciphertext-Policy Attribute-Based Encryption):

2.2.1. Tổng quan

CP-ABE là một phương pháp mã hóa tiên tiến, cho phép mã hóa dữ liệu dựa trên các thuộc tính của người dùng và các chính sách truy cập. Với scheme AC17 cho phép quản lý và kiểm soát truy cập dữ liệu một cách hiệu quả, đặc biệt hữu ích trong các hệ thống phân tán như điện toán đám mây:

- **Bảo mật:** CP-ABE AC17 cung cấp mức độ bảo mật cao bằng cách mã hóa dữ liệu dựa trên các thuộc tính và chính sách truy cập chi tiết.
- **Linh hoạt:** Sơ đồ hỗ trợ các chính sách truy cập phức tạp và động, cho phép quản lý truy cập dữ liệu một cách linh hoạt.
- **Hiệu suất:** AC17 cải thiện hiệu suất mã hóa và giải mã so với các sơ đồ trước đó, giúp tăng tốc độ xử lý và giảm tải tính toán.

Do đó, CP-ABE có nhiều ưu điểm hơn so với các kỹ thuật mã hóa khác. Nó cho phép chủ sở hữu dữ liệu mã hóa dữ liệu của họ theo cách cho phép kiểm soát truy cập chi tiết, giảm nguy cơ truy cập trái phép và vi phạm dữ liệu. Nó cũng cho phép chia sẻ dữ liệu một cách an toàn trên các hệ thống phân tán mà không cần cơ quan trung quản lý kiểm soát truy cập. Ngoài ra, CP-ABE có thể mang lại mức độ linh hoạt cao, cho phép chủ sở hữu dữ liệu điều chỉnh các chính sách và thuộc tính truy cập khi cần.

Tuy nhiên, CP-ABE cũng có một số hạn chế. Nó có thể tồn kén về mặt tính toán, đặc biệt khi xử lý lượng lớn dữ liệu và các chính sách truy cập phức tạp. Ngoài ra, có thể khó quản lý và duy trì các chính sách truy cập khi số lượng thuộc tính và người dùng tăng lên.

Nhìn chung, CP-ABE là một kỹ thuật mã hóa mạnh mẽ có thể cung cấp khả năng kiểm soát truy cập chi tiết và chia sẻ dữ liệu an toàn trong các hệ thống phân tán. Ưu điểm và hạn chế của nó cần được xem xét cẩn thận khi thực hiện chiến lược bảo mật dữ liệu

2.2.2. Về FAME: Fast Attribute-based Message Encryption

Mã hóa tin nhắn dựa trên thuộc tính nhanh hay FAME là một phương pháp mã hóa cho phép liên lạc an toàn giữa hai bên dựa trên chất lượng thay vì danh tính cụ thể. Đây là một kỹ thuật mã hóa dựa trên thuộc tính (ABE) nhằm mục đích hiệu quả và có thể mở rộng, cho phép mã hóa và giải mã tin nhắn nhanh chóng ngay cả trong các hệ thống quy mô lớn có nhiều người dùng.

FAME cung cấp một số lợi ích so với các hệ thống ABE khác, khiến nó trở thành một lựa chọn phổ biến để liên lạc an toàn.

- Trước hết, nó sử dụng mã hóa lai, cung cấp thời gian mã hóa và giải mã nhanh chóng, khiến nó phù hợp để sử dụng trong các bối cảnh có nguồn lực hạn chế, chẳng hạn như thiết bị di động và Internet of Things.

- Thứ hai, nó có chi phí tính toán thấp, có nghĩa là quá trình mã hóa và giải mã sử dụng ít tài nguyên máy tính hơn.

- Thứ ba, nó có cơ chế kiểm soát truy cập linh hoạt cho phép thực hiện các chính sách kiểm soát truy cập chi tiết dựa trên nhiều thuộc tính. Cuối cùng, nó được thiết kế để đảm bảo an toàn trước một loạt các cuộc tấn công, bao gồm cả các cuộc tấn công thông đồng.

Nhìn chung, FAME là một chương trình mã hóa mạnh mẽ và hiệu quả, cung cấp khả năng kiểm soát truy cập linh hoạt và bảo mật mạnh mẽ. Điều này làm cho nó trở thành lựa chọn phổ biến cho nhiều ứng dụng, bao gồm nhắn tin an toàn, kiểm soát truy cập và chia sẻ dữ liệu.

2.2.3. Minh họa thuật toán

2.2.3.1. Thiết lập

Ở bước này, ta sẽ tạo ra 2 khóa là Master Key (MK) và Public Key (PK).

```
def setup(cpabe, path):
    public_key, master_key = cpabe.ac17.setup()

    serialized_public_key = objectToBytes(public_key, cpabe.groupObj)
    serialized_master_key = objectToBytes(master_key, cpabe.groupObj)

    save_to_file(serialized_public_key, path+'public_key.bin')
    save_to_file(serialized_master_key, path+'master_key.bin')

    print(f"Keys generated and saved to {path}/public_key.bin and {path}/master_key.bin")
```

Hình 6: Tạo Master Key và Public Key

2.2.3.2. Tạo Secret Key

Dựa vào Master Key (MK) đã tạo được ở bước trên cùng với các tập thuộc tính của người dùng (DU), sẽ tạo ra khóa bí mật Secret Key (SK) của mỗi người dùng.

```
def gen_secret_key(cpabe, public_key_file, master_key_file, attributes, private_key_file):
    public_key = bytesToObject(load_from_file(public_key_file), cpabe.groupObj)
    master_key = bytesToObject(load_from_file(master_key_file), cpabe.groupObj)

    user_attributes = attributes.split(',')
    private_key = cpabe.ac17.keygen(public_key, master_key, user_attributes)

    serialized_private_key = objectToBytes(private_key, cpabe.groupObj)
    save_to_file(serialized_private_key, private_key_file)

    print(f"Secret Key generated and saved to {private_key_file}")
```

Hình 7: Tạo Secret Key

2.2.3.3. Mã hóa

Sau khi đã có đầy đủ các thành phần, sẽ bắt đầu tiến hành mã hóa với các tham số đầu vào là: Public Key (PK), Message (M) và Policy (P).

Trong đó:

- **Public Key (PK):** Có được từ bước thiết lập.

- **Message (M):** Trong trường hợp tổng quát thì sẽ là thông điệp cần mã hóa. Tuy nhiên, đối với nhóm chúng em, đây sẽ là một file .csv bao gồm (columns, keys) với column là tên các cột trong Database và keys là khóa của AES-GCM khi mã hóa các cột.

- **Policy (P):** Đây là chính sách truy cập. Việc mã hóa dựa vào chính sách truy cập để khi giải mã, chỉ những người có thuộc tính thỏa mãn thì mới có thể giải mã được thông điệp.

Sau khi mã hóa, chúng em sẽ đưa bản mã của các khóa AES-GCM lên Cloud (do chưa tìm được cách padding các khóa vào column của Database trước đó, nên chúng em sẽ tạo một bảng trong Database đó để lưu các keys).

```
def AC17encrypt(self, public_key, message, policy):
    random_key = self.groupObj.random(GT)

    # Encrypt random_key using CP-ABE
    encrypted_key = self.ac17.encrypt(public_key, random_key, policy)
    # Serialize to save to database
    encrypted_key_b = self.serialized.jsonify_ctxt(encrypted_key)
    # Create key for AES by random_key
    hash = hashlib.sha256(str(random_key).encode())
    key = hash.digest()
    aes = AES.new(key, AES.MODE_GCM)

    if type(message) != bytes:
        if type(message) != str:
            message = str(message)

    ciphertext, authTag = aes.encrypt_and_digest(message.encode())
    nonce = aes.nonce

    # Final ciphertext that will be sent to database
    ciphertext = nonce + ciphertext + authTag

    len_encrypted_data = len(encrypted_key_b)
    encrypted_data = len_encrypted_data.to_bytes(8, byteorder='big') + encrypted_key_b.encode() + ciphertext
    # Encode Base64 for encrypted_key and ciphertext

    encrypted_data = base64.b64encode(encrypted_data).decode()
    return encrypted_data
```

Hình 8: Tạo bảng để lưu Keys

2.2.3.4. Giải mã

Việc giải mã của CP-ABE yêu cầu đầu vào là bản mã (CT), khóa Secret Key (SK) và các thuộc tính của người dùng (A).

Trong đó:

- **Bản mã (CT):** có được từ việc mã hóa các khóa AES-GCM bằng CP-ABE. Chúng em phải lấy bảng mã này từ Cloud về để giải mã chúng.
- **Khóa Secret Key (SK):** có được từ việc tạo Secret Key ở bước 2.
- **Tập thuộc tính người dùng (A):** Bao gồm các thuộc tính tương ứng với người dùng đó, được thiết lập sẵn. Khi giải mã sẽ lần lượt dựa trên các thuộc tính đó để xem người dùng có giải mã dữ liệu hay không. Ví dụ: ['TEACHER', 'DEAN', 'UIT'].

```
def AC17decrypt(self, public_key, encrypted_data, private_key):  
    encrypted_data = base64.b64decode(encrypted_data.encode())  
    len_encrypted_key = int.from_bytes(encrypted_data[:8], byteorder='big')  
    encrypted_key_b = encrypted_data[8:8 + len_encrypted_key]  
    ciphertext = encrypted_data[8 + len_encrypted_key:]  
  
    encrypted_key = self.serialized.unjsonify_ctxt(encrypted_key_b.decode('utf-8'))  
    recovered_random_key = self.ac17.decrypt(public_key, encrypted_key, private_key)  
  
    if recovered_random_key:  
        nonce = ciphertext[:16]  
        authTag = ciphertext[-16:]  
        ciphertext = ciphertext[16:-16]  
  
        hash = hashlib.sha256(str(recovered_random_key).encode())  
        key = hash.digest()  
        try:  
            aes = AES.new(key, AES.MODE_GCM, nonce)  
            recovered_message = aes.decrypt_and_verify(ciphertext, authTag)  
            return recovered_message.decode()  
        except ValueError as e:  
            return None  
    else:  
        return None
```

Hình 9: Giải mã CP-ABE

2.3. ABAC (Attribute-Based Access Control)

ABAC là một mô hình kiểm soát truy cập hiện đại, linh hoạt, dựa trên các thuộc tính (attributes) của các thực thể như người dùng, tài nguyên, hành động và ngữ cảnh. ABAC cho phép các tổ chức xác định và thi hành các chính sách truy cập chi tiết và phức tạp, giúp bảo vệ tài nguyên một cách hiệu quả và an toàn hơn. Nhóm chúng em quyết định sử dụng ABAC để kiểm soát xem ai có quyền truy cập Cloud Database.

2.3.1. Xác định các thuộc tính

Xác định các thuộc tính liên quan đến người dùng, tài nguyên, hành động và ngữ cảnh. Các thuộc tính này có thể bao gồm:

- **Người dùng:** Vai trò, cấp bậc, nhóm, địa chỉ IP.
- **Tài nguyên:** Loại tài nguyên, nhãn độ nhạy, chủ sở hữu.
- **Hành động:** Loại hành động (đọc, viết, xóa), phương thức truy cập.
- **Ngữ cảnh:** Thời gian truy cập, vị trí truy cập, điều kiện mạng.

2.3.2. Xác định các chính sách

Xây dựng các chính sách truy cập dựa trên các thuộc tính đã xác định. Một chính sách truy cập thường bao gồm:

- **Điều kiện (conditions):** Các biểu thức logic kết hợp các thuộc tính.
- **Quyết định (decision):** Cho phép hoặc từ chối truy cập nếu các điều kiện thỏa mãn.

2.3.3. Thi hành chính sách

Khi có yêu cầu truy cập, hệ thống ABAC sẽ:

- Thu thập các thuộc tính của người dùng, tài nguyên, hành động và ngữ cảnh của họ.
- Đánh giá các thuộc tính này dựa trên các chính sách truy cập đã xác định.
- Đưa ra quyết định cho phép hoặc từ chối truy cập dựa trên kết quả đã đánh giá.

```

1  #policy_definitions.py
2  master_doctor_policy_json = {
3      "uid": "1",
4      "description": "Head of Cardiology department can view all documents of their department.",
5      "effect": "allow",
6      "rules": {
7          "subject": {
8              "$.role": {
9                  "condition": "Equals",
10                 "value": "head-of-cardiology"
11             },
12             "$.department": {
13                 "condition": "Exists",
14             },
15             "$.position": {
16                 "condition": "Equals",
17                 "value": "doctor"
18             },
19         },
20
21         "resource": {
22             "$.type": {
23                 "condition": "Exists",
24             }
25         },
26         "action": {
27             "$.method": {
28                 "condition": "Equals",
29                 "value": "upload"
30             }
31         },
32         "context": {}
33     },
34     "targets": {},
35     "priority": 1
36 }
37

```

Hình 10: Một policy được thiết lập sẵn cho ABAC

- Hình trên là một đoạn ABAC nhóm chúng em sử dụng để xây dựng cho hệ thống.

CHƯƠNG 3. TÀI NGUYÊN CỦA MÔ HÌNH

3.1. Công cụ và tài nguyên

3.1.1. Ngôn ngữ lập trình Python 3.10.11

Python là một ngôn ngữ lập trình bậc cao, được thiết kế với cú pháp đơn giản và dễ đọc, giúp lập trình viên viết mã nhanh chóng và hiệu quả. Python là một ngôn ngữ thông dịch, đa nền tảng, và có một kho thư viện phong phú. Các ưu điểm của Python bao gồm dễ học, năng suất cao, và cộng đồng mạnh mẽ, khiến nó trở thành lựa chọn lý tưởng cho nhiều lĩnh vực như phát triển web, khoa học dữ liệu, và trí tuệ nhân tạo.

3.1.2. Môi trường triển khai

- Center of Authority:

- + Thiết bị: Laptop Acer Swift 3 SF314-511-55QE Intel Core-i5 1135G7.
- + Hệ điều hành: Ubuntu 22.04 Jammy Jellyfish.

- Data Owner:

- + Thiết bị: Macbook Air 2019 Intel Core-i5 8210Y.
- + Hệ điều hành: macOS 14.5 Sonoma.

- Data User:

- + Thiết bị: Laptop Asus Zenbook 14 Oled Intel Core-i5 1240P.
- + Hệ điều hành: Ubuntu 22.04 Jammy Jellyfish.

3.1.3. Thư viện hỗ trợ

3.1.3.1. Charm-Crypto

Charm-Crypto là một framework mã nguồn mở dành cho nghiên cứu và phát triển các giao thức mật mã. Nó được thiết kế để dễ dàng mô phỏng và triển khai các thuật toán mật mã phức tạp, hỗ trợ nhiều loại nhóm mật mã và cung cấp các công cụ để nhanh chóng triển khai các thuật toán mới. Charm-Crypto hỗ trợ nhiều giao thức mật mã bao gồm các giao thức dựa trên cặp (pairing-based) và mật mã đường cong elliptic (elliptic curve cryptography), và là một công cụ quan trọng cho các nhà nghiên cứu trong lĩnh vực mật mã.

3.1.3.2. PyCryptodome

PyCryptodome là một thư viện mã nguồn mở cho Python, được thiết kế để thay thế PyCrypto với các cải tiến về bảo mật và chức năng. Thư viện này cung cấp các công cụ mạnh mẽ để thực hiện các thao tác mật mã như mã hóa, giải mã, băm (hashing), và tạo chữ ký số. PyCryptodome có API đơn giản và dễ hiểu, hỗ trợ nhiều thuật toán mật mã hiện đại như AES, RSA, SHA-256, và được tối ưu hóa để cung cấp hiệu năng cao trong các ứng dụng thực tế, đảm bảo an toàn trước các lỗ hổng bảo mật mới.

3.1.3.3. OpenSSL

OpenSSL là một thư viện phần mềm mạnh mẽ và mã nguồn mở, cung cấp các công cụ và giao thức cần thiết để thực hiện các chức năng bảo mật như mã hóa, giải mã, và chứng thực trên Internet. Được phát triển nhằm hỗ trợ các giao thức TLS (Transport Layer Security) và SSL (Secure Sockets Layer), OpenSSL cung cấp các thư viện mạnh mẽ cho các ngôn ngữ lập trình như C và C++, cùng với các công cụ dòng lệnh để tạo và quản lý các chứng chỉ số, khóa riêng, và các yếu tố bảo mật khác. OpenSSL được sử dụng rộng rãi trong nhiều ứng dụng và dịch vụ trực tuyến để bảo vệ dữ liệu và đảm bảo sự riêng tư và toàn vẹn của các kết nối mạng.

3.1.3.4. PBC (Pairing-Based Cryptography)

PBC là một thư viện mã nguồn mở được thiết kế để hỗ trợ nghiên cứu và triển khai các thuật toán mật mã dựa trên cặp (pairing-based cryptography). Được phát triển bởi Ben Lynn tại Đại học Stanford, PBC cung cấp các công cụ cần thiết để làm việc với các phép toán cặp đôi trên các nhóm elliptic. Thư viện này rất hữu ích trong việc xây dựng các giao thức mật mã phức tạp như mã hóa dựa trên thuộc tính (attribute-based encryption), chữ ký ngưỡng (threshold signatures), và hệ thống chứng minh không tương tác (non-interactive zero-knowledge proofs). PBC hỗ trợ nhiều loại cặp và cấu trúc nhóm khác nhau, giúp các nhà nghiên cứu và nhà phát triển thử nghiệm và triển khai các giải pháp mật mã tiên tiến một cách hiệu quả.

3.1.3.5. GMP (GNU Multiple Precision)

Thư viện GNU Multiple Precision (GMP) là một thư viện miễn phí dành cho các phép tính số học có độ chính xác tùy ý, hoạt động trên các số nguyên, số hữu tỉ và số dấu phẩy động. GMP được thiết kế để đạt hiệu suất cao trong các ứng dụng yêu cầu độ chính xác cao, chẳng hạn như mật mã học, đại số tính toán và lý thuyết số. GMP cung

cấp một bộ hàm phong phú cho các phép tính số học, so sánh và thao tác trên các số lớn, và được tối ưu hóa mạnh mẽ để đạt tốc độ cao. Thư viện này được viết bằng ngôn ngữ C và hỗ trợ nhiều nền tảng và kiến trúc khác nhau.

3.1.3.6. Py-ABAC

Py-ABAC là một thư viện Python mã nguồn mở được sử dụng để triển khai kiểm soát truy cập dựa trên thuộc tính (Attribute-Based Access Control - ABAC). Trong mô hình ABAC, quyền truy cập được quyết định dựa trên các thuộc tính của người dùng, tài nguyên, và môi trường thay vì các quyền tĩnh. Py-ABAC cung cấp các công cụ để định nghĩa, quản lý và thực thi các chính sách truy cập phức tạp dựa trên các thuộc tính này. Thư viện hỗ trợ viết chính sách bằng JSON, cho phép kiểm tra quyền truy cập linh hoạt và mạnh mẽ theo các quy tắc kinh doanh cụ thể. Py-ABAC được sử dụng rộng rãi trong các ứng dụng cần kiểm soát truy cập chi tiết và linh hoạt, đảm bảo rằng chỉ những người dùng có các thuộc tính phù hợp mới có thể truy cập vào các tài nguyên nhất định.

3.1.3.7. Bcrypt

Bcrypt là một thư viện mã hóa mật khẩu được thiết kế để bảo vệ mật khẩu bằng cách sử dụng một thuật toán băm với độ phức tạp cao. Bcrypt kết hợp hàm băm Blowfish với một yếu tố bảo mật gọi là "salt" và khả năng điều chỉnh độ khó để đảm bảo rằng ngay cả khi mật khẩu bị đánh cắp, việc giải mã sẽ cực kỳ khó khăn. Nó làm cho các cuộc tấn công bẻ khóa mật khẩu, chẳng hạn như brute-force, trở nên không hiệu quả và tốn nhiều thời gian. Bcrypt được đánh giá cao vì khả năng bảo vệ mật khẩu mạnh mẽ và được sử dụng rộng rãi trong các ứng dụng web và dịch vụ trực tuyến để đảm bảo an toàn cho thông tin người dùng.

3.1.3.8. PyQt6

PyQt6 là một bộ công cụ phát triển giao diện người dùng (GUI) cho Python, dựa trên thư viện Qt phiên bản 6. PyQt6 cung cấp các lớp và công cụ cần thiết để xây dựng các ứng dụng desktop phức tạp và đa nền tảng với giao diện đẹp mắt và hiệu suất cao. PyQt6 hỗ trợ tất cả các tính năng mạnh mẽ của Qt6, bao gồm quản lý cửa sổ, xử lý sự kiện, vẽ đồ họa, và nhiều widget khác nhau. Nó cũng cung cấp tích hợp chặt chẽ với các tính năng của hệ điều hành và hỗ trợ các mô hình lập trình hướng đối tượng, chức năng, và tín hiệu-khe (signal-slot) của Qt. PyQt6 rất phổ biến trong cộng đồng phát triển phần mềm nhờ tính dễ sử dụng, tài liệu phong phú, và khả năng mở rộng linh hoạt.

3.1.4. Cơ sở dữ liệu

3.1.4.1. MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở phổ biến, được phát triển bởi MySQL AB và hiện nay thuộc sở hữu của Oracle Corporation. MySQL sử dụng ngôn ngữ truy vấn cấu trúc (SQL) để quản lý và thao tác dữ liệu trong các bảng. Được biết đến với hiệu suất cao, độ tin cậy và dễ sử dụng, MySQL là lựa chọn hàng đầu cho nhiều ứng dụng web và doanh nghiệp.

Nó hỗ trợ nhiều tính năng mạnh mẽ như giao dịch, khóa ngoại, chỉ mục toàn văn (full-text indexing), và sao lưu (replication), làm cho nó trở thành một giải pháp linh hoạt và mạnh mẽ cho các nhu cầu quản lý dữ liệu.

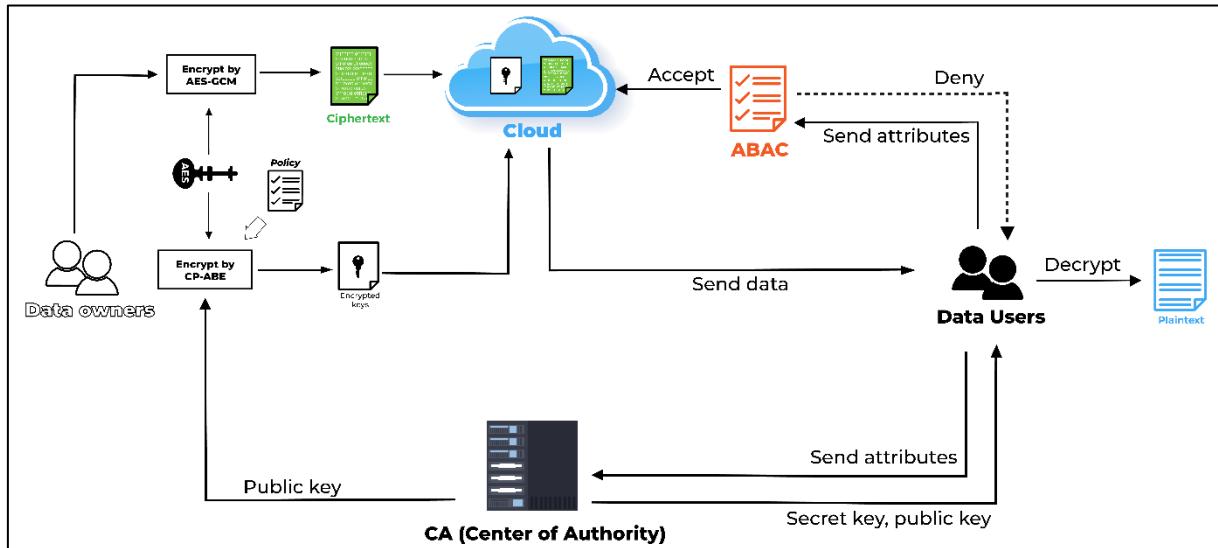
3.1.4.2. MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL mã nguồn mở, được thiết kế để lưu trữ và quản lý dữ liệu không có cấu trúc. Thay vì sử dụng các bảng và hàng như trong cơ sở dữ liệu quan hệ, MongoDB sử dụng các tài liệu dạng JSON (JavaScript Object Notation) với cấu trúc linh hoạt, cho phép lưu trữ dữ liệu phức tạp và có thể thay đổi theo thời gian. MongoDB hỗ trợ khả năng mở rộng ngang (horizontal scaling) thông qua sharding, giúp quản lý dữ liệu lớn hiệu quả.

Ngoài ra, MongoDB cung cấp các tính năng như chỉ mục đa dạng, sao lưu (replication), và khả năng truy vấn mạnh mẽ, phù hợp cho nhiều ứng dụng web hiện đại và các hệ thống quản lý dữ liệu lớn.

3.2. Kiến trúc hệ thống

3.2.1. Tổng quan kiến trúc hệ thống



Hình 11: Tổng quan kiến trúc hệ thống

Kiến trúc hệ thống trên gồm 4 nodes: Data Owners, Cloud, CA, Data Users.

- **Data Owners:** Các trưởng khoa bệnh viện, người sẽ cập nhật thêm bệnh nhân và tình hình sức khỏe của các bệnh nhân đến khám.
- **Cloud:** Dịch vụ AWS RDS và chỉ dùng để lưu trữ và quản lý dữ liệu.
- **Center of Authority:** Cung cấp public key cho Data Owners để mã hóa, cung cấp secret key và public key cho Data Users để giải mã (dựa theo thuộc tính của họ).
- **Data Users:** Các bác sĩ, y tá, người muốn truy xuất dữ liệu để xem thông tin và tình hình sức khỏe của bệnh nhân đến khám.

3.2.2. CA (Center of Authority):

Server có khả năng thiết lập và quản lý các kết nối SSL/TLS với các client, và thực hiện các chức năng như tạo Secret key và gửi Public Key. Nhóm chúng em triển khai CA thông qua class Server dưới đây:

```
class Server:  
    def __init__(self, host='127.0.0.1', port=10023, certfile=None,  
keyfile=None):  
        self.host = host  
        self.port = port  
        self.certfile = certfile  
        self.keyfile = keyfile  
        self.is_running = True  
        self.cpabe = CPABE("AC17")
```

```

        # self.setup_key("setup/") # Automatically run setup_key during
initialization

def TAssetup(self, path):
    try:
        setup(self.cpabe, path)
        print('Setup successfully completed.')
    except Exception as e:
        print(f"Error during setup: {e}")

def TAGenkey(self, conn, addr, msg, public_key_file, master_key_file,
private_key_file_path):
    try:
        mode, attributes = msg.split('|', 1) # Chỉ tách lần đầu tiên
tìm thấy '|'
        print(f"Mode: {mode}, Attributes: {attributes}")
        send_attributes = attributes.upper()
        print(f"Attributes upper: {send_attributes}")

        if mode == 'genkey':
            # Tạo khóa bí mật
            gen_secret_key(self.cpabe, public_key_file, master_key_file,
send_attributes, private_key_file_path)

            # Đọc nội dung của file khóa bí mật
            with open(private_key_file_path, 'rb') as private_key_file:
                private_key = private_key_file.read()
                conn.sendall(private_key + b'END_OF_FILE') # Gửi khóa
bí mật đến client

            print('Generated secret key for client:', addr)
    except Exception as e:
        print(f"Error generating secret key for {addr}: {e}")
    finally:
        # Xóa file khóa bí mật sau khi đã gửi
        if os.path.exists(private_key_file_path):
            os.remove(private_key_file_path)
        conn.close()

def TASendPubKey(self, conn, addr, public_key_file):
    try:
        # Đọc nội dung của file khóa công khai
        with open(public_key_file, 'rb') as public_key_file:
            public_key = public_key_file.read()
            conn.sendall(public_key + b'END_OF_FILE') # Gửi khóa công
khai đến client

        print('Send public key for client:', addr)
    except Exception as e:
        print(f"Error sending public key to {addr}: {e}")

```

```

        finally:
            conn.close()

    def handle_request(self, conn, msg, addr):
        try:
            if msg == 'setup':
                self.TASetup("setup/")
            elif msg.startswith('genkey|'):
                self.TAGenkey(conn, addr, msg, "setup/public_key.bin",
                            "setup/master_key.bin", "setup/private_key.bin")
            elif msg == 'get_pub_key':
                self.TASendPubKey(conn, addr, "setup/public_key.bin")
            else:
                conn.sendall('Invalid choice'.encode('utf-8'))
                print(f"Invalid choice from {addr}")
        except Exception as e:
            print(f"Error handling request from {addr}: {e}")
        finally:
            conn.close()

    def start(self):
        server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        server_socket.bind((self.host, self.port))
        server_socket.listen(5)
        print(f"Server listening on {self.host}:{self.port}")

        context = ssl.create_default_context(ssl.Purpose.CLIENT_AUTH)
        context.load_cert_chain(certfile=self.certfile,
                               keyfile=self.keyfile)

        while self.is_running:
            try:
                conn, addr = server_socket.accept()
                conn = context.wrap_socket(conn, server_side=True)
                msg = conn.recv(1024).decode()
                self.handle_request(conn, msg, addr)
            except Exception as e:
                print(f"Error accepting connection: {e}")
                if conn:
                    conn.close()

        server_socket.close()

    def setup_key(self, path):
        self.TASetup(path)

if __name__ == "__main__":
    server = Server(certfile='localhost.crt', keyfile='localhost.key')
    server.start()

```

CHƯƠNG 4. TRIỂN KHAI

4.1. Kịch bản triển khai

4.1.1. Data Owners

- Xây dựng một Cloud Server Database trên Amazon RDS MySQL để làm nơi lưu trữ bản mã của dữ liệu và bản mã của các khóa AES.

- Bcrypt sẽ là lớp xác thực mật khẩu đầu tiên của Data Owners (và Data Users) trước khi vào hệ thống.

- Sau khi vào được hệ thống, Data Owners sẽ tiến hành thao tác chọn dữ liệu cần mã hóa và mã hóa chúng. Sau đó Data Owners sẽ tiếp tục mã hóa các khóa của AES-GCM có được từ lần mã hóa trước đó bằng CP-ABE.

- Từ những dữ liệu đã được mã hóa, Data Owner (cũng) tiến hành thiết lập những quyền hạn truy cập cho Data Users thông qua ABAC. Sau cùng, sẽ đưa toàn bộ dữ liệu lên Cloud và lưu trong các bảng khác nhau. Tất cả đều được thực hiện tự động sau khi Data Owners hoàn tất thao tác chọn dữ liệu đưa lên Cloud.

4.1.2. Data Users

- Data Users ở một máy khác sẽ tiến hành đăng nhập vào hệ thống và cũng sẽ thông qua Bcrypt để xác thực mật khẩu. Tiếp đến, Data User sẽ được kiểm tra quyền truy cập của mình vào loại dữ liệu mà Data User mong muốn thông qua ABAC, nếu đủ điều kiện thì có thể tiếp tục truy cập vào để lấy dữ liệu từ Cloud.

- Sau đó, Data Users sẽ thực hiện các thao kiểm tra đã có Secret Key (SK) chưa. Nếu chưa sẽ gửi yêu cầu đến CA (qua kết nối TLS/SSL) để lấy Public Key (PK) và Secret Key (SK) để tiến hành tải bản mã của khóa AES-GCM từ Cloud về máy và giải mã chúng. Dựa vào thuộc tính của họ, CA sẽ tạo ra Secret Key và trả về cho Data User. Sau đó, dựa vào thuộc tính của họ, Data User sẽ được phép lấy số cột dữ liệu tương ứng với chính sách mà Data Owner đã thiết lập trước đó.

- Sau khi giải bản mã và có được khóa AES-GCM, Data User sẽ tiếp tục thực hiện lấy bản mã của dữ liệu từ Cloud về máy và giải mã số cột tương ứng với số lượng khóa AES có được.

4.1.3. CA (Center of Authority):

Được triển khai trên 1 máy riêng (Server) và luôn bật để chịu trách nhiệm cung cấp các khóa cho các bên khác. TA cung cấp public key cho Data Owners để mã hóa dữ liệu và tạo secret key cho Data Users để giải mã dữ liệu trong hệ thống.

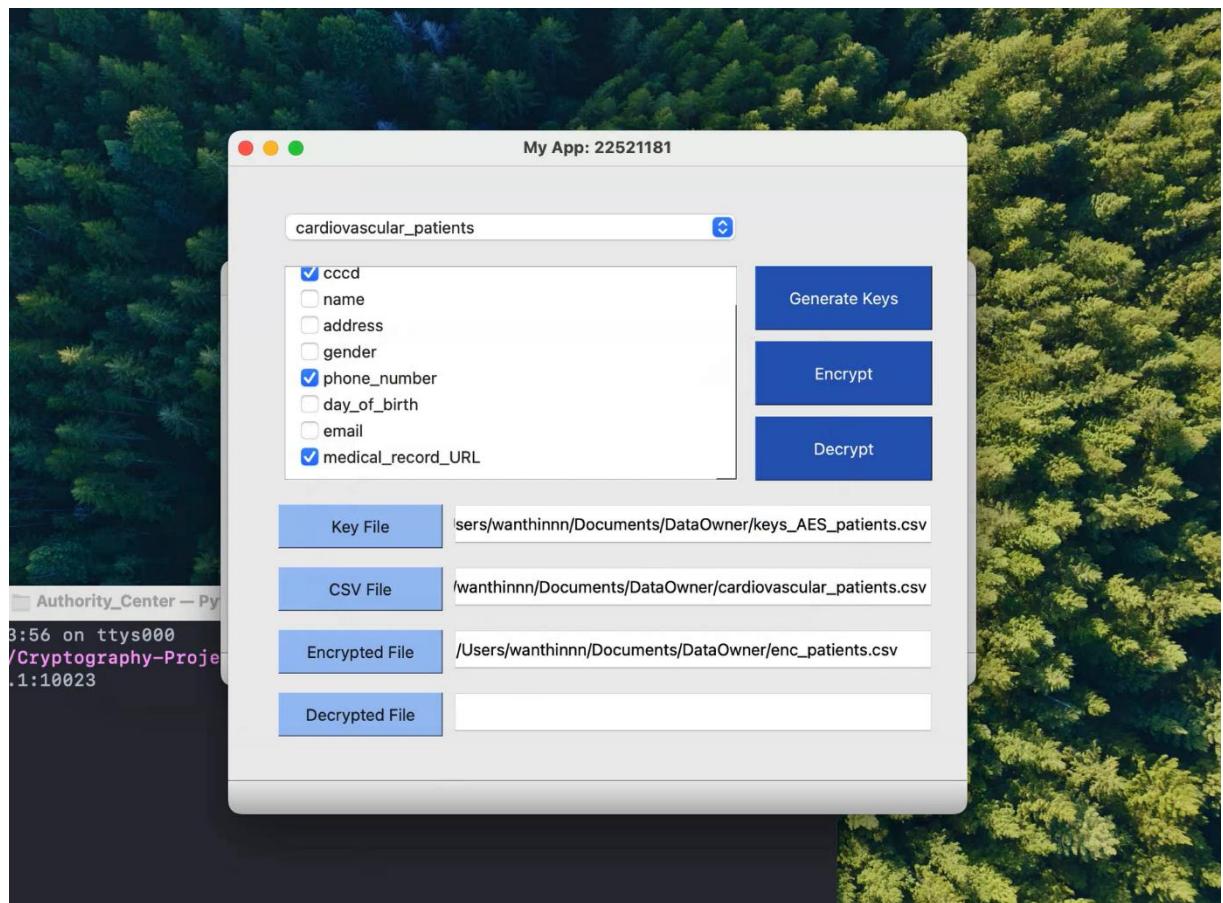
4.2. Vấn đề bảo mật và giải pháp của mô hình:

Thông qua đây, chúng em nhận thấy rằng mỗi thao tác thực hiện từ việc đăng nhập, lấy, gửi các khóa cần thiết cho quá trình mã hóa, giải mã hay lấy, đưa dữ liệu với Cloud đều rất quan trọng trong việc bảo mật và đảm bảo tính toàn vẹn dữ liệu. Chính vì thế, việc xây dựng các kênh truyền an toàn thông qua TLS/SSL trong mạng nội bộ sẽ giúp chúng em hạn chế được vấn đề trên.

4.3. Kiểm thử và kết quả kiểm thử:

4.3.1. Data Owners

4.3.1.1. Mã hóa



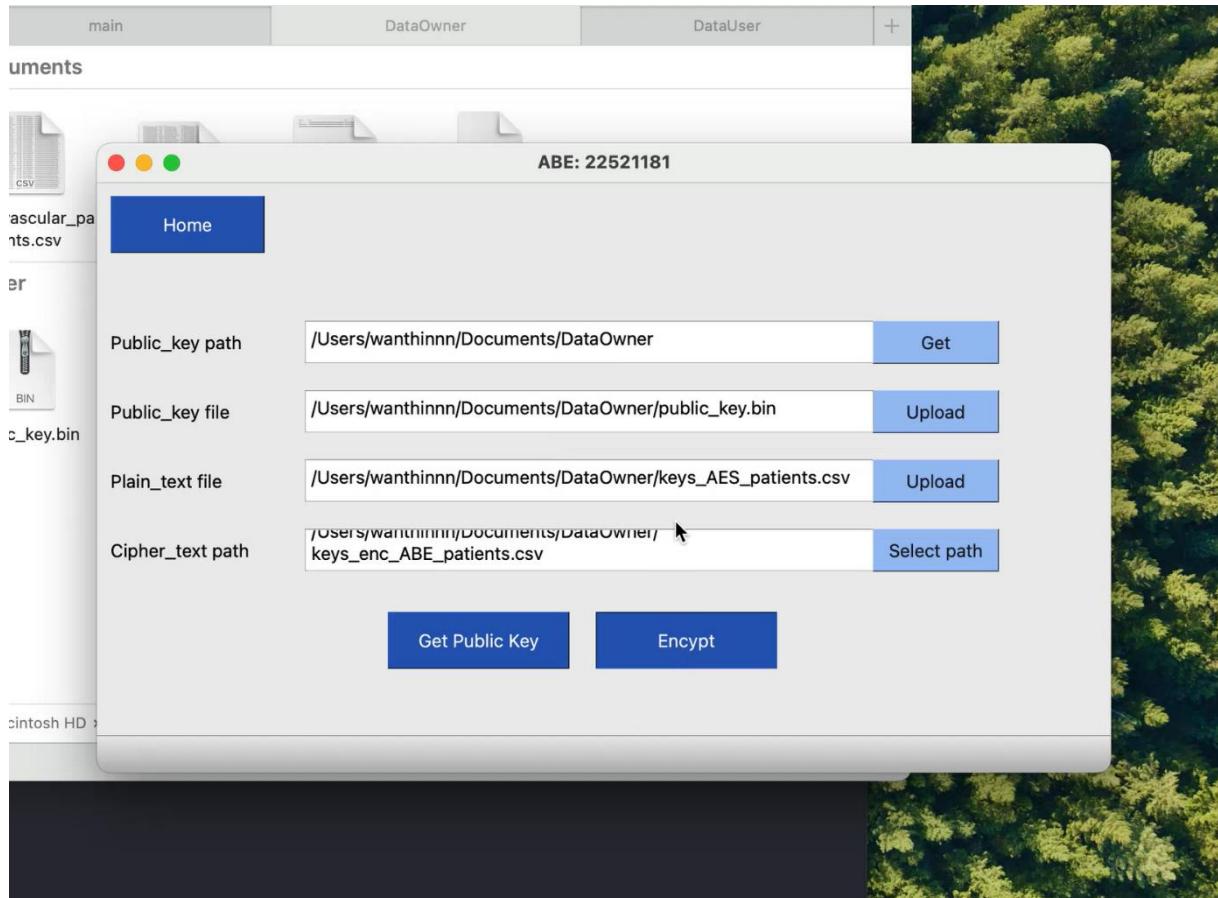
Hình 12: Triển khai mã hóa dữ liệu

- Owners sẽ chọn Database để mã hóa bằng AES-GCM.

- Ở đây, nhóm chúng em sẽ mã hóa 3 cột là “cccd”, “, phone_number” và “medical_record_URL”.

id	cccd	name	address	gender	phone_number	day_of_birth	email
KH001	Z9NQVUMuAK3spZ/pL2VY1n+UvKjZE7+6oISNlw==	Oliver Smith	15 Le Loi Street, District 1, Ho Chi Minh City	Female	R9xM1WYy+mX/dRSmRjp2PVGzG7THpg==	15/01/1980	oliversmith203@gmail.com
KH002	Z9pUUcIC6lqqjnyldD6kMjD1ho68NhiihQa==	Sophia Johnson	75 Nguyen Hue Street, District 1, Ho Chi Minh City	Male	R91IIK7oyIY9VYMaFuymWJL6Frvwx0OQ==	22/03/1995	sophiajohnson306@gmail.com
KH003	adisV80lBKjps5vn3BTVVyAz8t5kNp0PxMrw==	Liam Brown	12 Tran Hung Dao Street, Hoan Kiem District, Hanoi	Female	R95kJFCvayezZBPG/Z27QGV/S6uuoMR0mw==	08/11/1974	liambrown172@gmail.com
KH004	ZdpVUcwgBa/zoZrqNhgns8D20vorNPOJD63VkQ==	Isabella Davis	25 Vo Van Tan Street, District 3, Ho Chi Minh City	Female	R99jJFCBuaQpjt/1r1Wc9e3yCmlxtPg==	30/06/1988	isabelladavis276@gmail.com
KH005	YNNSVMYIBKrpqzpz5/q5OSc3IL1LMdrVely9A==	Elijah Miller	56 Hai Ba Trung Street, District 1, Ho Chi Minh City	Female	R9HJ1GUx++RID+cPd16t7akUN31T6Q==	19/02/1990	elijahmiller819@gmail.com
KH006	Zt5SWsIuBk7pqZf/o/jw47BY3+f+voc18TPIXhA==	Mia Wilson	68 Ly Tu Trong Street, District 1, Ho Chi Minh City	Male	R9jhJ6Vu6SyWLeBfGeQkmQOrPFH86Q==	27/09/1983	miawilson885@gmail.com
KH007	YttxVsUjBqjtj7p7QBwhr3g+2wzYhYrDZD2h6q==	James Moore	34 Nguyen Du Street, District 1, Ho Chi Minh City	Male	R9pgKv+c2z+2ThwTH8Yb/0CQx9M14Yjw==	13/04/1977	jamesmoore603@gmail.com
KH008	YNxeV8cuBksop/nlpJhz+xYr8tDvM7qygewA==	Amelia Taylor	40 Nguyen Thi Minh Khai Street, District 1, Ho Chi Minh City	Female	R9tVKFadzOyUz/zQSrovqODYwajGc+jHAw==	05/12/1992	ameliataylor126@gmail.com
KH009	ZdxWLUseKc67jObhtnywNgkpd038QW6R4k9JA==	William Anderson	22 Nguyen Dinh Chieu Street, District 1, Ho Chi Minh City	Male	R9RuJveezUvtPhOB8aWsr09Q/070QzA==	18/08/1985	williamanderson216@gmail.com
KH010	Y9teUmojBq7r05bv1b2WFT2dUkGZ+r37ZoxT5g==	Harper Thomas	29 Tran Phu Street, District 5, Ho Chi Minh City	Male	R9VnIFSyuqW02gBB7RdZlyVBy76qMKwgg==	10/07/1978	harperthomas609@gmail.co
KH011	Y9NVUsMhAriqj3sL0xMBtyletq5SeLSmd9Sw==	Benjamin Johnson	88 Pasteur Street, District 1, Ho Chi Minh City	Male	R9VnIFSyuqW02gBB7RdZlyVBy76qMKwgg==	02/05/1984	benjaminjohnson206@gmail.com
KH012	adtrUcIkAazzop/c09/rLdvzAx49dAxrxwsQ==	Evelyn White	91 Le Thanh Ton Street, District 1, Ho Chi Minh City	Male	R91kJFCvayezCzo1tskpv/Sy8kYB8fQ==	11/03/1999	evelynwhite673@gmail.com
KH013	Z9tTU8jAkruoSmCD7D/9QYK9tLkZt0U4p	Lucas Harris	55 Ba Huyen Thanh Quan Street, District 3, Ho Chi Minh City	Female	R95pJFCvxaQfNIz4DNBOoJdFM1T1PA==	23/01/1981	lucasharris133@gmail.com
KH014	ZNSWV8wuBK/j5p3RsZ9wMtpaqo2f576Pg==	Abigail Martin	64 Truong Dinh Street, District 3, Ho Chi Minh City	Male	R99jJGUx++R91+acOCRH3e3gPPFoK6w==	14/12/1976	abigailmartin473@gmail.com
KH015	Y9tWBYk863pZjsj774swf7qgyNXLcUwde0A==	Henry Thompson	101 Tran Quang Kai Street, District 1, Ho Chi Minh City	Male	R9rhj6VwzuS44QSYSpJCCxP51XBjw==	09/06/1987	henrythompson379@gmail.com
KH016	Zd9SVUyNbh6rvcZrtVTKLUSzH7F1PoAtBE2gQ==	Ella Garcia	120 Nam Ky Khoi Nghia Street, District 1, Ho Chi Minh City	Female	R9gkVv+c2z+2T+c2s/3Gf7Koefd14EQ==	24/04/1994	ellagarica903@gmail.com
KH017	Z9xWWSMuCqypppvjGislr1BzDfUp5qYw==	Alexander Martinez	45 Le Van Sy Street, District 3, Ho Chi Minh City	Female	R9pkFadzOy5R0k/4e+3cPtttE6cht0==	31/10/1982	alexandermartinez655@gmail.com
KH018	YthXUsUc6Vrpp7pacHvRkXBWZpaqpmaggLYqw==	Avery Robinson	78 Nguyen Trai Street, District 1, Ho Chi Minh City	Male	R9tuJveezuVf19zmyrIE/0vB1X8AvJfw==	06/11/1991	averyrobinson629@gmail.com
KH019	NNXV8UjBqBergZ3nEQtRIVEWJG12DX7SrSnw==	Michael Clark	130 Nguyen Tri Phuong Street, District 5, Ho Chi Minh City	Male	R9RnIFSyuqW+Y7bHonTOuCeFkyIAwN==	07/08/1975	michaelclark321@gmail.com
KH020	ZNpWUMAIcq3opZrpCgthsZMhJkfEZXDAGe2Q5w==	Scarlett Rodriguez	37 Bui Vien Street, District 1, Ho Chi Minh City	Female	R9Vm1WYy+mXQgRYsnWfEb0010xebTuQ==	28/02/2000	scarlettrodriguez204@gmail.com
KH021	Y91SU8iAVgtppjnxw6+tQjgMhWjz/CXNzQv==	Daniel Lewis	92 Dinh Tien Hoang Street, District 1, Ho Chi Minh City	Female	R9xkJFCvayezMd5yz++d+E+JzfUyUAFY59w==	29/05/1989	daniellew1432@gmail.com
KH022	Z9pFvB8yAqFspj3r6Loif131p+jjAfe9E90Ew==	Grace Lee	103 Doan Van Bo Street, District 4, Ho Chi Minh City	Female	R91jJFCbxaQa/LSMZf45xwicea74kkUg==	15/03/1979	gracelee499@gmail.com
KH023	Z9hQuicB8k7uo5fqW2yMs1pWYk9VY8t8pU6w==	Ethan Walker	86 Cach Mang Thang Tam Street, District 3, Ho Chi Minh City	Female	R95jJGUx++R3mIAadpMVG4/vBRdsuEXQ==	25/12/1986	ethanwalker872@gmail.com
KH024	ZtRUcYBqrgp21Cb6AWs+HL28adeDMdJZ	Chloe Hall	27 Nguyen Van Cu Street, District 5, Ho Chi Minh City	Male	R99jJ6Vwze6SNzQULprjt/xV8lufcerXQ==	17/09/1993	chloehall756@gmail.com
KH025	Y1tQUslkC6grpbvzOpHkg2b0p9CaNyZvg==	Matthew Allen	39 Nguyen Huu Canh Street, Binh Thanh District, Ho Chi Minh City	Female	R9gkVv+c2z+2T0szw/JHRX4sPQFBcwp==	21/01/1978	matthewallen670@gmail.com
KH026	ZdVJUs0jBK3rJbnSCflit1BSOnPRg16wuciA==	Victoria Young	63 Dien Bien Phu Street, District 1, Ho Chi Minh City	Male	R9vKfadzOyUmjFk1Me5OrP2P0/6s3AKbw==	01/08/1985	victoriayoung653@gmail.com
KH027	aN9UJcwAqD/pzqNchx91OMf2kwyIdgR+5N	Samuel Hernandez	51 Ton That Tung Street, District 1, Ho Chi Minh City	Male	R9puJveezUvbDg4eTtLj2GKElyQ==	13/11/1980	samuelhernandez738@gmail.com
KH028	YN5TVCclBa7jZ2fmQbT7Y8M40hbuXLtvqAQ==	Riley King	19 Pham Ngu Lao Street, District 1, Ho Chi Minh City	Female	R9nIFSyuqWuAgNmNUKQzvd/3M6zb0hw==	04/04/1976	rileyking550@gmail.com
KH029	7thWjBYB6mts53tkeGo3nOLnR4dBeGU+uMFA==	David Wright	73 Tran Binh Trong Street, District 5, Ho Chi Minh City	Female	R9Bm1WYy+mXoKCIUoR7t9irAvGfVw==	22/06/1998	davidwright577@gmail.com

Hình 13: Dữ liệu sau khi được mã hóa



Hình 14: Mã hóa Keys AES bằng CP-ABE

- Sau đó, tiếp tục mã hóa các khóa AES-GCM bằng CP-ABE với chính sách được cài đặt sẵn cho từng khóa.

columns	key
cccd	AAAAAAAAABaJ7InBvbGljeS16ICloKEhFQUQtT0YtQ0FSREIPTe9HWSBhbmQgQ0FSREIPTe9HWSkgYW5kIERPQ1RPUiLCAiQ18wIjogWyiyOkkkWJGJcW1lc29UMUtWOEMyTvpcQXUJry9Db3NeAFM1RmxwUJ
phone_number	AAAAAAAAABu7InBvbGljeS16ICloKChlRUFEU9GLUNBUKRJTxPR1kgj3lgTIVSU0UpIgfzCDBDQVJESU9MT0dZKSbhbhMqgRE90VE9SKSIsICJDxzlObIbjI6bTBt3UwQm90KzVKUVZ2T2zSi85d2pkb21HTV
medical_record_URL	AAAAAAAABGR7InBvbGljeS16ICloTVSU0gb3lgSEVRC1PRI1DQVJESU9MTdZKSIsICJDxzlObIbjI6U3FrkF2Z0SHVkmilveZXY1paNG90N3cvHJtODhXrnZCSUYyT3dGRUQwcFe0WIRCDlhGSIVQr

Hình 15: Các Keys AES sau khi được mã hóa

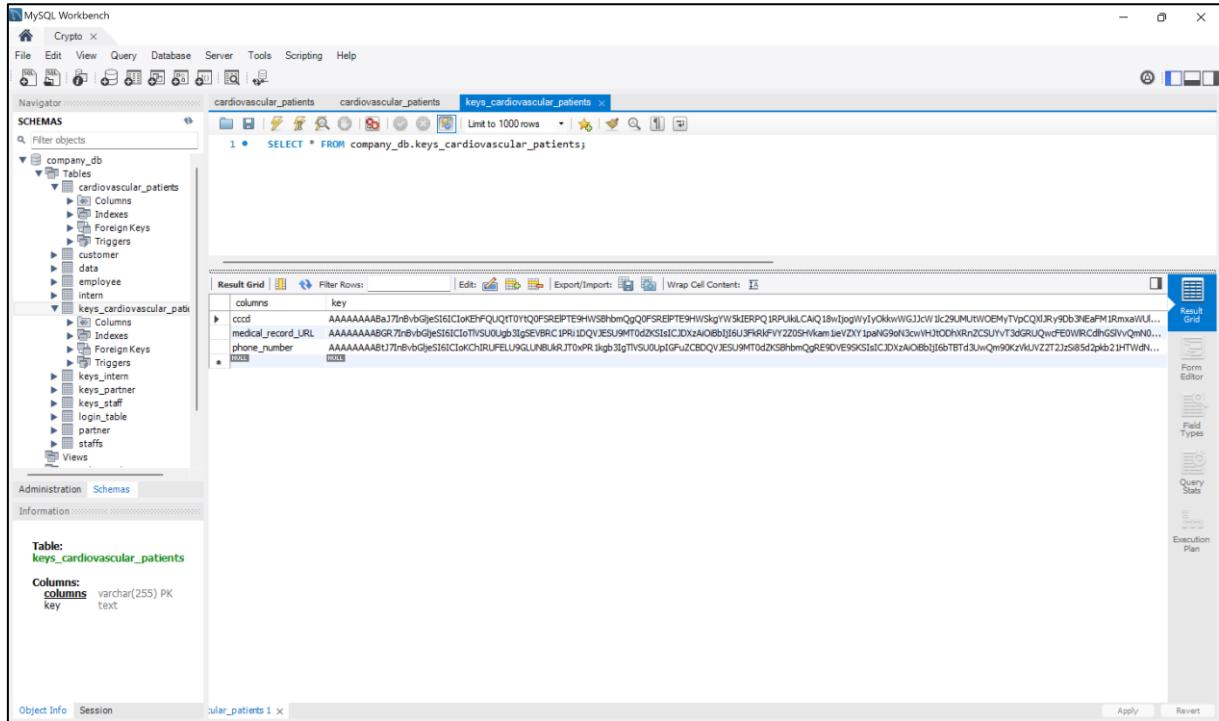
- Các khóa AES-GCM sau khi được mã hóa bằng CP-ABE.

4.3.1.2. Đưa dữ liệu lên Cloud

cardiovascular_patients										
id	cccd	name	address	gender	phone_number	day_of_birth	email	medical_record_URL	Result Grid	Form Editor
KH001	Z9hQVbMLunAK3spZ/pL2V7Y	Oliver Smith	15 Le Loi Street, District 1, Ho Chi Minh City	Female	R9m11WYy+mrXfR	15/01/1980	oliversmith203@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH002	ZSpuJLclCqjQj3yjD	Sophia Johnson	75 Nguyen Hue Street, District 1, Ho Chi Minh City	Male	R9j8K2Qy9Y9y	22/03/1995	sophiajohnson305@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH003	ad5f808b945f545b78	Liam Brown	12 Tran Hung Dao Street, Hoan Kiem District, H...	Female	R9k5IV0yayeeZBP	08/11/1974	lambrown172@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH004	ZdpfVLvngBqjqp2nrlhlg	Isabella Davis	25 Vo Van Tan Street, District 3, Ho Chi Minh City	Female	R99j3FCpduQoPjt	30/06/1988	isabeladavis276@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH005	YN1uSWMyRkoppo5+jn9h	Elijah Miller	56 Hai Ba Trung Street, District 1, Ho Chi Minh City	Female	R9h11GQx+4RfD+	19/02/1990	elijahmiller819@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH006	ZTS5Wiu1u8p2p2fjy4u	Mia Wilson	68 Ly Tu Trong Street, District 1, Ho Chi Minh City	Male	R9h36VwzsuSy	27/09/1983	miaowlow85@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH007	Yt0tVslqllgt017p7Q9j	James Moore	34 Nguyen Du Street, District 1, Ho Chi Minh City	Male	R9pgKvH+z+27Hw	13/04/1977	jamesmoore603@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH008	YNxeviCuKvsoopInp2jh	Amelia Taylor	40 Nguyen Thi Minh Khai Street, District 1, Ho Ci...	Female	R9t9KFpdoyJzJ	05/12/1992	ameliataylor126@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH009	ZdvfUUsC5j7p3bnymj	William Anderson	22 Nguyen Dinh Chieu Street, District 1, Ho Chi ...	Male	R9t9UlfeweuPtm	18/08/1985	williamanderson216@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH010	Y9tehJMojs7p7o5vBzW	Harper Thomas	29 Tran Phu Street, District 5, Ho Chi Minh City	Male	R9t9IF5FyuuQw0g	10/07/1978	harperthomas609@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH011	Y9hVJUmh7nAqrcj3cLox	Benjamin Jackson	88 Pasteur Street, District 1, Ho Chi Minh ...	Male	R9t9JQy0Qy0/37	02/05/1984	benjaminjackson206@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH012	ad9fUclkaazosp09/r	Evelyn White	91 Le Thanh Ton Street, District 1, Ho Chi Minh ...	Male	R9t9JVCwzG	11/03/1999	evelynwhite73@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH013	29jTU8lgKuox5p/CD7/	Laura Horne	55 Be Huyen Thanh Quan Street, District 3, Ho ...	Female	R95j3FCbxwsPN	23/01/1981	laurahorne133@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH014	Zt8hVivwvldt/jp8p3kQd	Alaina Marin	64 Truong Dinh Street, District 3, Ho Chi Minh City	Male	R99j1GQx+4RfD+	14/10/1976	alainamarin473@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH015	19jW81k63sqzjh774	Henry Thompson	101 Tran Quang Kha Street, District 1, Ho Ci...	Male	R9h36Vzus644Q	09/06/1987	henrythompson179@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH016	2695Mrh686qzq7vT1KL	Ella Garcia	120 Nam Ky Khoi Nghie Street, District 1, Ho Chi ...	Female	R9t9KvH+z+27Hw	24/04/1994	ellagarcia903@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH017	29xWWm4kCqppovVjyG	Alexander Martin	45 Le Van Sy Street, District 3, Ho Chi Minh City	Female	R9p9KFpdoySRu	31/10/1982	alexandermartine255@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH018	Yth1UsJuc6Vpp/b4hV	Avery Robinson	78 Nguyen Trai Street, District 1, Ho Chi Minh City	Female	R9t9UvezeeuV19	06/11/1991	averyrobinson629@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH019	aN9JUWUlgBarp23nCQ8	Michael Clark	39 Nguyen Tri Phuong Street, District 5, Ho Ci...	Male	R9t9IF5FyuuQw0g	07/08/1975	michaelclark221@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH020	ZtpnULM4Cq3opJpCg	Scarlett Rodriguez	37 Buuien Street, District 1, Ho Chi Minh City	Female	R9t9JWVyyhM0QGr	28/02/2000	scarletrodiguez204@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH021	ZdPfV8p1Afp7pppnw6	Daniel Lewis	92 Dinh Tien Hoang Street, District 1, Ho Chi Mi...	Female	R9t9JWVyyhM0QGr	29/03/1989	daniellewis432@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH022	Z9pV8TA4Ps9j3y61L0	Grace Lee	103 Doan Van Bo Street, District 4, Ho Chi Min...	Female	R95j3FCbxwsPN	15/03/1979	gracelee499@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH023	Z9hQLMgBKJu5f9V2m	Ethan Walker	86 Cach Mang Thang Tam Street, District 3, Ho ...	Male	R95j1GQx+4Rf3m	25/12/1986	ethanwalker872@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH024	ZtR9Uk7BrgpZ1Xb6A	Chloe Hall	27 Nguyen Van Cu Street, District 5, Ho Chi Min...	Male	R99j36Vzus644Q	17/09/1993	chloehall756@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH025	Yt1Qu8dc6ypbvxxOP	Matthew Allen	39 Nguyen Huu Canh Street, Binh Thanh District...	Female	R9t9KvH+z+27Hw	21/01/1978	matthewallen670@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH026	ZdflUpaBj3pJbnSCf	Victoria Young	63 Dien Bien Phu Street, District 1, Ho Chi Min...	Male	R9t9KFpdoyJzJ	01/08/1985	victoriayoung559@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH027	aN9JUlwvAgJp2qzOtv9	Samuel Hernandez	51 Ton That Thung Street, District 1, Ho Chi Min...	Male	R9t9UvezeeuV1bM	13/11/1980	samuehernandez738@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH028	YNSTV0cB7qJzqQn9	Riley King	19 Pham Ngu Lao Street, District 1, Ho Chi Min...	Female	R9t9IF5FyuuQw0g	04/04/1976	rileyking550@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		
KH029	ZthWU8Y6n53kEg	David Wright	73 Tran Hung Trong Street, District 5, Ho Chi Min...	Female	R9t9JWVyyhM0QGr	22/06/1998	davidwright577@gmail.com	Jhr3K5YNgB318eRFPbodCBhFV91ZnWFgb0D...		

Hình 16: Đưa dữ liệu đã mã hóa lên Cloud.

- Bản mã của dữ liệu được đưa lên Cloud.

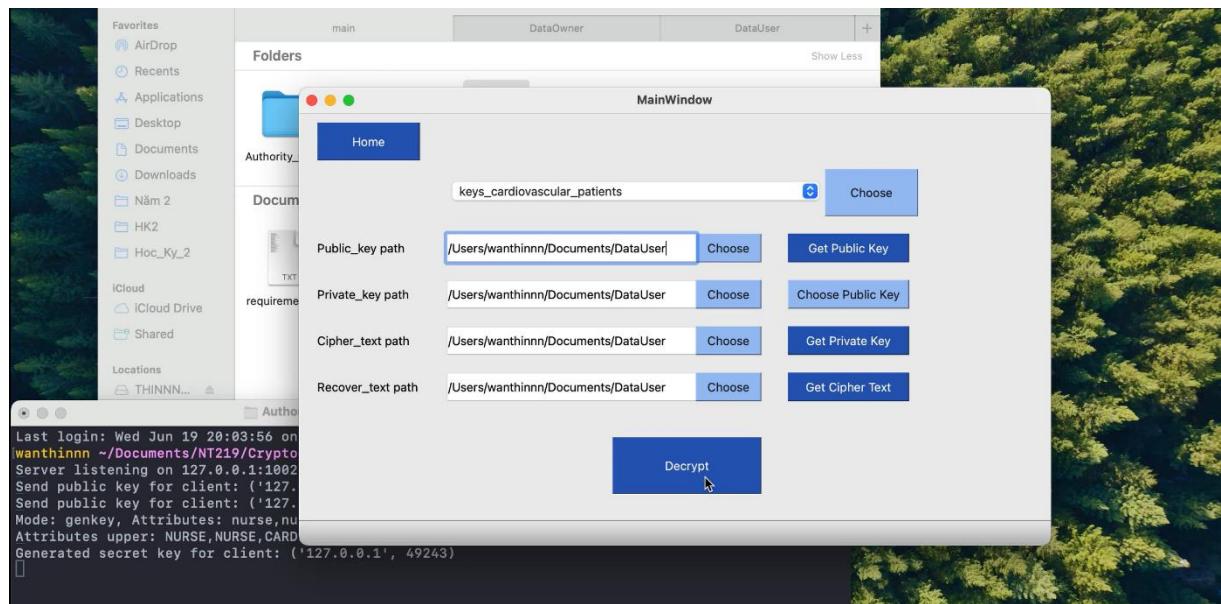


Hình 17: Đưa các Keys AES đã mã hóa lên Cloud.

- Bản mã của các khóa AES-GCM được đưa lên Cloud.

4.3.2. Data Users

4.3.2.1. Giải mã khóa AES-GCM thông qua CP-ABE



Hình 18: Yêu cầu Keys theo thuộc tính

- Sau khi User đăng nhập vào hệ thống, nếu chưa có Public Key (PK) và Secret Key (SK) thì sẽ gửi kết nối đến Server để trả về các keys theo thuộc tính của mình.

- Tiếp tục lấy các khóa AES-GCM từ Cloud về máy để giải mã sau khi đã có đủ key cần thiết.

```

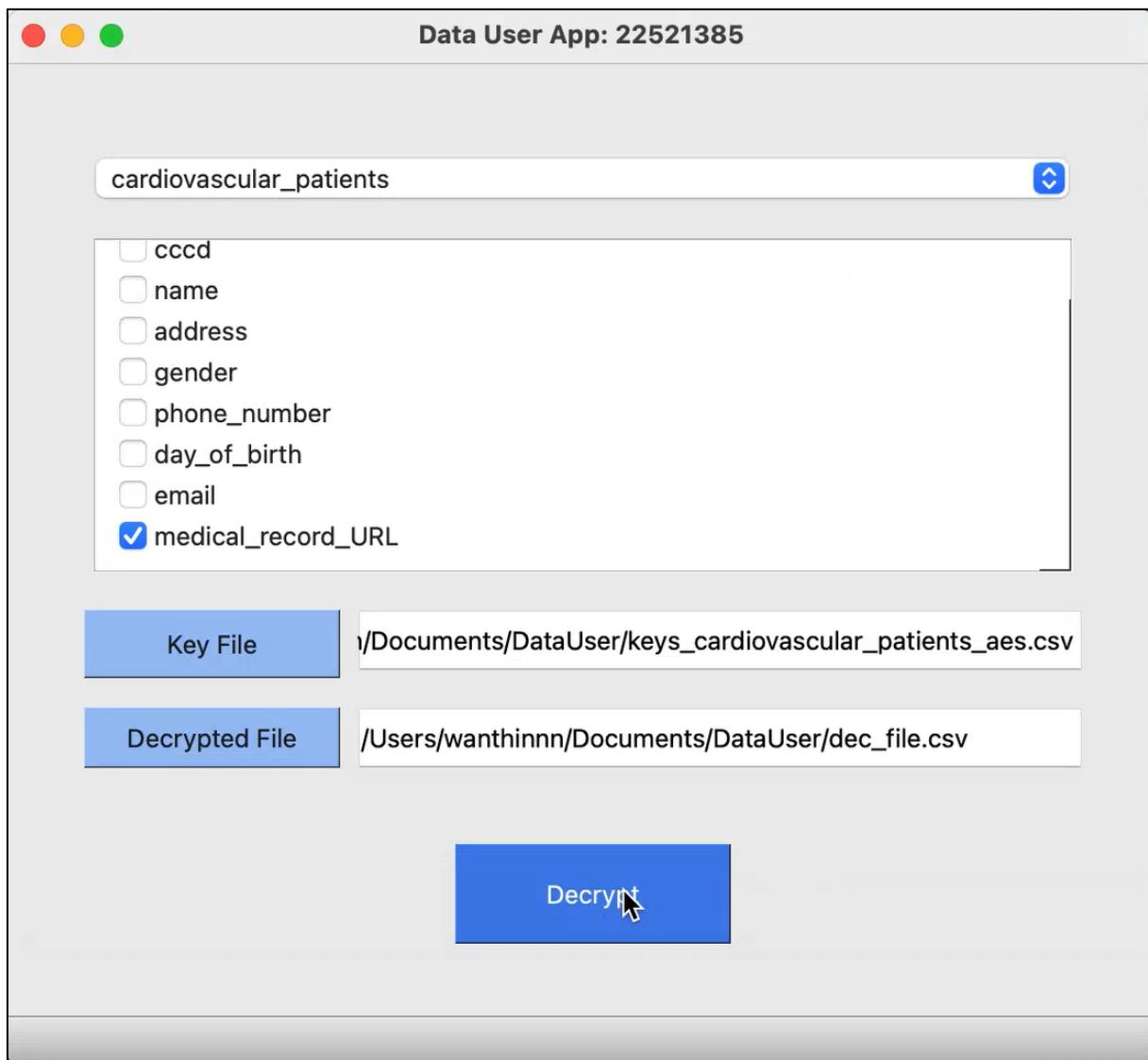
keys_cardiovascular_patients_aes.csv > data
1 columns, key
2 medical_record_URL, y6c133TURzx9f/w6mSoE6wTuUvTGMgTSGsNItaKs2x2kYc+pZRps9UlMhT4=
3

```

Hình 19: Thực hiện giải mã các Keys nhận được

- Đôi với User này chỉ có thuộc tính là “NURSE”, chỉ được phép xem hồ sơ bệnh án của bệnh nhân, nên khi giải mã khóa AES-GCM chỉ trả về khóa của cột “medical_record_URL”.

4.3.2.2. Giải mã dữ liệu thông qua AES-GCM



Hình 20: Giải mã dữ liệu thông qua AES-GCM-256

- Sau khi có key AES-GCM của cột “medical_record_URL”, ta sẽ tiến hành giải mã cột này để lấy dữ liệu cần thiết.

```

id,cccd,name,address,gender,phone_number,day_of_birth,email,medical_record_URL
KH001,29QVQVMUAK3ap/pl2YV,Oliver Smith,"15 Le Loi Street, District 1, Ho Chi Minh City",Female,R9xmIIWyy+mX/dr,15/01/1980,oliversmith203@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH002,29pUuc1Gl6qJjnvlD,Sophia Johnson,"75 Nguyen Huu Street, District 1, Ho Chi Minh City",Male,R91lLkZyOiy9VV,22/03/1995,sophiajohnson306@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH003,ad1SW881Bkjnpw3Bp/wn3BP,Isabella Davis,"12 Tran Hung Dao Street, Hoan Kiem District, Hanoi",Female,R95kJWoyeeZZBp,08/11/1974,Liambrown172@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH004,2dpUvcwBg/aoNrzNhg,Isabella Davis,"25 Vo Van Tan Street, District 3, Ho Chi Minh City",Female,R99jJFCbxuoQjt,30/06/1988,isabelladavis276@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH005,YNTSMYIBKrcopzps/5,Ellijah Miller,"51 Hai Ba Trung Street, District 1, Ho Chi Minh City",Female,R91hJL6Vzu6S4R,19/02/1990,elijahmiller819@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH006,2tSSksuI8K7pnqfjw4,Mia Wilson,"68 Ly Tu Trong Street, District 1, Ho Chi Minh City",Male,R95kJFCbxuoQjt,27/09/1983,miawilson85@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH007,YttXVsul1Rqjto7p7Q8w,James Moore,"34 Nguyen Du Street, District 1, Ho Chi Minh City",Female,R99jJFCbxuoQjt,13/04/1977,jamesmoore603@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH008,Ykx8e8uKvaop/nipJh,Amelia Taylor,"40 Nguyen Thi Minh Khai Street, District 1, Ho Chi Minh City",Female,R87vKFadzOyUz/z,05/12/1992,ameliataylor126@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH009,IdnHua8KC67jcbntryM,William Anderson,"22 Nguyen Dinh Chieu Street, District 1, Ho Chi Minh City",Male,R9pRuIVeezeuVpm,18/08/1985,williamanderson216@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH010,Y7ceuM0Bjg7r05vb1zW,Harper Thomas,"29 Tran Phu Street, District 5, Ho Chi Minh City",Male,R95hF5Fyudq029,10/07/1978,harperthomas609@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH011,Y9NVuS0h4q1q/s3L0x,Benjamin Jackson,"88 Pasteur Street, District 1, Ho Chi Minh City",Male,R9xLlkZyOiy9VV,02/05/1984,benjaminjackson206@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH012,adtrUc1Kaxosop/r09/r,Evelyn White,"91 Lu Thanh Ton Street, District 1, Ho Chi Minh City",Male,R91kJWoyeeZGt,11/03/1998,ewlynwhite573@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH013,29lTU81gKruo5mD70,Lucas Harris,"55 Ba Huyen Thanh Quan Street, District 3, Ho Chi Minh City",Female,R95kJFCbxuoQfN,23/01/1981,lucas'harris133@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH014,ZNSW8w8uBk/jp5zp3Rz,Abigail Martin,"64 Truong Dinh Street, District 3, Ho Chi Minh City",Male,R99jJFCbxuoQo/l,14/12/1976,abigailmartin473@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH015,Y97fw8YkB63aqjsh74,Henry Thompson,"101 Tran Quang Khanh Street, District 1, Ho Chi Minh City",Male,R99hJL6Vzu6S44C,09/06/1987,henrythompson379@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH016,29pSUMyBGrvqzrtVTKL,Ella Garcia,"128 Nam Ky Khoi Nghia Street, District 1, Ho Chi Minh City",Female,R99pKFadzOyUStC,24/04/1998,ellegarcia903@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH017,YthXusulC6vrpp7pachV,Avery Robinson,"45 Le Van Sy Street, District 3, Ho Chi Minh City",Female,R9pukFadzOyUSR,31/10/1982,alexandermartinez55@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH018,YthXusulC6vrpp7pachV,Avery Robinson,"28 Nguyen Trai Street, District 1, Ho Chi Minh City",Male,R91kJWoyeeVf9,06/11/1991,averyrobinson629@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH019,ANIXV8U8gBarq3nEQR,Michael Clark,"130 Nguyn Tri Phung Street, District 5, Ho Chi Minh City",Male,R99nIFsyuqWY7,07/08/1975,michaelclar321@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH020,2pnhuMALc3opzPzCgt,Scarlett Rodriguez,"37 Bui Vien Street, District 1, Ho Chi Minh City",Female,R99mT1My+mXQgR,28/02/2000,scarlettrodriqe284@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH021,Y91SU81v6Gptppjsnmw6,Daniel Lewis,"92 Dinh Tien Hoang Street, District 1, Ho Chi Minh City",Female,R9XkWJQoyeeZhds,29/05/1989,daniellewis423@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH022,29pF8Y1aQspzQ8y6Ylo,Grace Lee,"103 Doan Van Bo Street, District 4, Ho Chi Minh City",Female,R99jJFCbxuoQo/l,15/03/1979,gracelee499@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH023,19hQuCkgB7rno5fqW2Y,Ethan Walker,"86 Cach Hang Thang Tam Street, District 3, Ho Chi Minh City",Female,R95fJ16uX++R3m,25/12/1986,ethanwalker872@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH024,2t1RucyJBrqnp13Cb6A,Chloe Hall,"27 Nguyen Van Cu Street, District 5, Ho Chi Minh City",Male,R99hJL6Vzu6NzQ,17/09/1993,clhoehall1756@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH025,Yt1QUs1kccjprpbvxDP,Matthew Allen,"39 Nguyen Huu Canh Street, Binh Thanh District, Ho Chi Minh City",Female,R95hJL6Vzu6SHd,06/11/1978,matthewallen78@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH026,2d1Vu5oJBk3npbnScF1,Victoria Young,"63 Dien Bien Phu Street, District 1, Ho Chi Minh City",Male,R9LkVkfDzOyUmf,01/08/1985,victoriayoung65@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH027,AN9UUCwAg/pzqnp0hXc,Samuel Hernandez,"51 Ton That Tung Street, District 1, Ho Chi Minh City",Male,R99pIVeezeuVbm,13/03/1980,samuelhernandez738@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH028,VNTVcc1ba7jzofmfQb9,Riley King,"19 Phan Ngu Lao Street, District 1, Ho Chi Minh City",Female,R9tnIFsyuqWuq,04/04/1974,rileyking55@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH029,27hNU8YB6ntp53tkedG,David Wright,"73 Tran Dien Trong Street, District 5, Ho Chi Minh City",Female,R99hJL6Vzu6S44C,22/06/1989,davidwright577@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH030,29pVWMMUIC67rq3uobwg,Aria Lopez,"84 Nguyen Dinh Chieu Street, District 3, Ho Chi Minh City",Male,R99vLILkYy+o1YGD,03/07/1977,arielopez489@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH031,29pVWMMUIC67rq3uobwg,Joseph Scott,"48 Bui Th Xuan Street, District 1, Ho Chi Minh City",Female,R99jJFCbxuoQQt+,16/10/1989,josephscott329@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH032,YHTRUcu1Ba7opfpuVv6,Lily Green,"95 Tran Quoc Thao Street, District 3, Ho Chi Minh City",Female,R99jJFCbxuoQo/l,08/05/1982,lilygreen393@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH033,Yp0LUUMVBya3ap5/nhF6,Sebastian Adams,"66 Cao Thang Street, District 3, Ho Chi Minh City",Female,R95hJL6Vzu6SHd,12/09/1994,sebastiandanams14@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH034,ad1Xu5s8gQvypzqEn/h,Nora Baker,"102 Ho Tng Mau Street, District 1, Ho Chi Minh City",Male,R99gKVy++zT8vL,30/03/1983,norabaker272@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH035,Y91XUcEgCg7r7q3kP7z,Jackson Gonzalez,"85 Nguyen Van Troi Street, Phu Nhuan District, Ho Chi Minh City",Female,R9hukFadzOyUsNv,27/12/1988,jacksongonzalez653@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH036,adpRUuXkBejv05/snRtd,Rey Nelson,"117 Phm Dinh Phung Street, Phu Nhuan District, Ho Chi Minh City",Female,R99nFsyuqKpl,06/02/1996,zoeynelson224@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH037,2t1Uo5oLBardppp14T,Levi Carter,"21 Nguyen Van Linh Street, District 7, Ho Chi Minh City",Female,R99nFsyuqKu7,24/11/1987,levicarter64@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH038,YNeve8EJqnq53q1Hd,Hannah Mitchell,"38 Ton Duc Thang Street, District 1, Ho Chi Minh City",Female,R99mT1My+mXKMr,18/04/1987,hannahmitchell186@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH039,adpUu8wB63u0zq2H44,Gabriel Perez,"46 Nguyen Thi Dieu Street, District 3, Ho Chi Minh City",Male,R99LlkZyOiy9VV,25/10/1990,gabriel.perez124@gmail.com,31@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH040,Yt1UUMK8rqrz7bXmk,Ellie Roberts,"104 Tran Hung Dao Street, District 1, Ho Chi Minh City",Male,R99WJQoyeeZg5,15/07/1992,elie_roberts_2@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH041,YtSeMMiAk3tOsnnmRMr,Owen Turner,"28 Nguyen Thien Thut Street, District 3, Ho Chi Minh City",Male,R9x1JGUx++R167,11/06/1981,owenturner326@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH042,29t1eu81K6v7qJfji,Addison Phillips,"32 Ly Tu Trong Street, District 1, Ho Chi Minh City",Female,R99hJL6Vzu6SyVm,13/02/1979,addisonphilips94@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH043,YN7SVAcAukvvoScalVL5c,Caleb Campbell,"59 Nguyen Dinh Chieu Street, District 3, Ho Chi Minh City",Male,R99gKVy++zT7t+,29/04/1986,calebcampbell1685@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH044,ZNkQV8uAjkpgo5ztSGfR,Zoe Parker,"89 Nguyen Trai Street, District 1, Ho Chi Minh City",Female,R99pKFadzOyUzg,19/12/1975,zoe parker_84@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH045,2d1eU81u6vtq7rDN8,Nathan Evans,"94 Vo Van Tan Street, District 3, Ho Chi Minh City",Female,R9huIVeezeuVAMQ,03/08/1988,nathanevans876@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details
KH046,ANvVU8wB6tqskrbv/,Stella Edwards,"44 Cao Thang Street, District 3, Ho Chi Minh City",Female,R9LkVkfDzOyUch,07/09/1993,stelawards175@gmail.com,https://vigorweb-storages.s3.amazonaws.com/details

```

Hình 21: Dữ liệu sau khi được giải mã

- Dữ liệu sau khi được giải mã bằng khóa AES-GCM có được.
- Cột “medical_record_URL” được giải mã, còn 2 cột “cccd” và “phone_number” không có khóa AES-GCM nên vẫn chỉ là bản mã như trên Cloud.

4.3.3. CA (*Center of Authority*)

4.3.3.1. Localhost

- Trong trường hợp trên, nhóm chúng em thực hiện demo trên một máy, do đó sẽ dùng địa chỉ IP và Port của Localhost.

- Certificate cho Server:

```
-----BEGIN CERTIFICATE-----
MIIC1zCCAb+gAwIBAgIUXOBtCebOibhxY3H0iP0xTY+pd08wDQYJKoZIhvcNAQEL
BQAwFDESMBAGA1UEAwJBG9jYWxob3N0MB4XDTI0MDYwNTIwMDg0M1oXDTI1MDYw
NTIwMDg0M1owFDESMBAGA1UEAwJBG9jYWxob3N0MIIBIjANBgkqhkiG9w0BAQE
AAOCAQ8AMIIBCgKCAQEAE4xq2HvrimRfb3Qh1x9Lp2Ae3rUR/ckIwyIL0aXpaRonf
b0mEpDyqTHCA9oQE00Qy6y6uDz9HMQdaSMB2cdBncGAqjRJRVeTP5VVYdxBQD4/7
7Qx5+GumG5FoA09ZMiIsNVWRE1JgI35BJQKtmU7XVvDPjRxQpu6+pTj1Y3K7JrC
0i0eAA09LBed7U4xdvcVKVMKK1bQnohA+zHT0tqvG3K9jBJJEgVH8GaX1AX9g1lw
iu+BX0+YMQW8IR402f/s9KZ7/lrQLAwKUyms+p7dIFGpFbHeDDzZoKCJpumHq6rn
DI15KeA3BF/M5yV7iPjIe99HdZuFX0dx2RG9iTJd0wIDAQABoyEwHzAdBgNVHQ4E
FgQUfkOoR7DFoESQ7ozKcMfvWUHWyG8wDQYJKoZIhvcNAQELBQADggEBAGYwv6UA
BqCkZ4+smew8hwdb00QPalHVhWku+LKZ/v1QSDoxlLdtYA+YlRQuq/vF6qOsXi0X
0WDWhJYRPx+G7FORnYctKEd/pYY+EMp2U+KUs+e1JF13h8ukBd4vns7MDe+MUvY1
Ry9JI/Mz2jqp3yZLIqwW1PJQBPCJmWqM2WAwhkegd3I6GSuLB8qzDiaCDHkXupt
V3uh2mZrLXUTSpSAsC6Cr0gqqDHWV31/CMEMwdsL+Vu9xqq9GCaye9cvnfCldFBr
d/jOSwuyePWXVKqq1uLEY8cTBdv13HZzzqTW6EBk8DLsj8RblCtZTTp4Tk7PGN0+
Y+XfWODZe71bS30=
-----END CERTIFICATE-----
```

Hình 22: Chứng chỉ cho Server (Localhost)

4.3.3.2. Client – Server

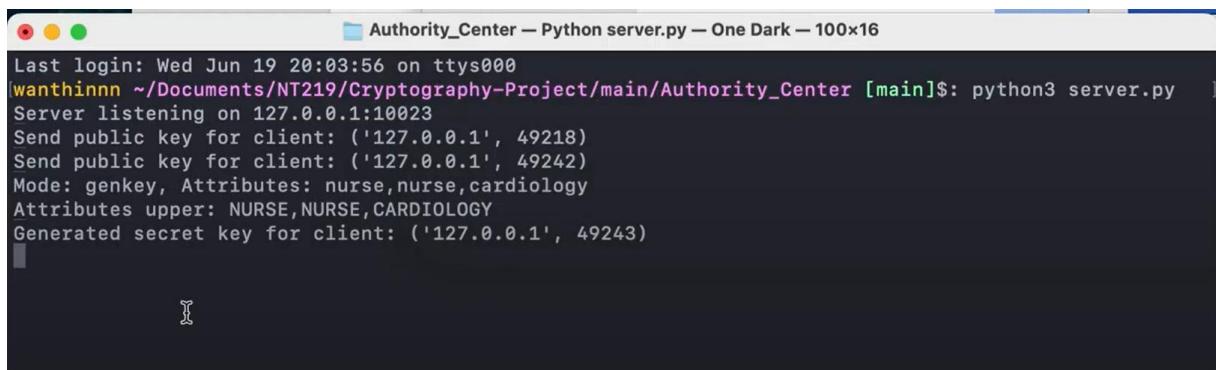
- Trong trường hợp này, Client – Server sẽ là các Data Users và CA, chúng được kết nối với nhau trong 1 mạng nội bộ của bệnh viện.

- Certificate của Server:

```
-----BEGIN CERTIFICATE-----  
MIIB7jCCAXQCFEVA7fBC1d0Vb5CdCvNfwj75+F5pMAoGCCqGSM49BAMCMFsxCzAJ  
BgNVBAYTAKFVMRMwEQYDVQQIDAپTb21lLVN0YXR1MSEwHwYDVQQKDBhJbnRlc  
dCBXaWRnaXRzIFB0eSBMdGQxFDASBgNVBAMMCzE5Mi4xNjguMS40MB4XDTI0MDYw  
OTIwMzAzNVoXTI1MDYwOTIwMzAzNVowWzELMAkGA1UEBhMCQVUxEzARBgNVBAgM  
ClNvbWUtU3RhdGUxITAfBgNVBAoMGE1udGVybмV0IFdpZGdpdHMgUHR5IEх0ZDEU  
MBIGA1UEAwLMTkyLjE2OC4xLjQwdjAQBgchкjOPQIBBгUrgQQAIgNiAAR2tnQZ  
0oP2Fx2pzTkHKz01x2YkwyuPKw0NABXmkP3jip79oRhbPU1sqRDEHFF8TKJkiuoI  
4Jo28u3N7LNU6iuWGs2A17ywF8IAT7UrU0tIsh5/QFbfxf28uZxxwrLPBG8wCgYI  
KoZIzj0EAwIDaAAwZQIwaqJUMX8zJrlqtU05wr7lt4RDnUIYJiMsfQgaBF+tizZV  
aBCvu3z0IoIrFayAPvPZAjEA4kBvGBp6iM/BTbWqDx5ISrZeRfaaNcWD4SWQa4uh  
11Qr6TGQHXMgy/r1oTMzLJqV  
-----END CERTIFICATE-----
```

Hình 23: Chứng chỉ cho Server

4.3.3.3. Gửi và nhận Public Key, Secret Key



```
Authority_Center — Python server.py — One Dark — 100x16  
Last login: Wed Jun 19 20:03:56 on ttys000  
wanthinnn ~/Documents/NT219/Cryptography-Project/main/Authority_Center [main]$: python3 server.py  
Server listening on 127.0.0.1:10023  
Send public key for client: ('127.0.0.1', 49218)  
Send public key for client: ('127.0.0.1', 49242)  
Mode: genkey, Attributes: nurse,nurse,cardiology  
Attributes upper: NURSE,NURSE,CARDIOLOGY  
Generated secret key for client: ('127.0.0.1', 49243)
```

Hình 24: Gửi và nhận Public Key, Secret Key

CHƯƠNG 5. TỔNG KẾT

5.1. Kết quả đạt được

Có 3 mục tiêu chúng em đã đề cập ở phần 1.4

1. Tính bảo mật - Đảm bảo rằng dữ liệu nhạy cảm được lưu trữ trên đám mây được bảo vệ khỏi sự truy cập hoặc tiết lộ trái phép.

2. Tính toàn vẹn - Đảm bảo rằng dữ liệu được lưu trữ trên đám mây không bị giả mạo hoặc sửa đổi trái phép.

3. Ủy quyền - Đảm bảo rằng người dùng truy cập dữ liệu dựa trên đám mây có mức độ quyền và đặc quyền truy cập phù hợp. Chắc chắn. Giải pháp của chúng tôi đã đạt được thành công ba mục tiêu chính đã được đặt ra.

Đầu tiên, chúng em đã triển khai một hệ thống xác thực người dùng bằng cách sử dụng Center of Authority (CA) và ABAC. Cách tiếp cận này đảm bảo rằng chỉ những người dùng được xác thực mới có thể truy cập vào hệ thống, bổ sung thêm một lớp bảo mật để bảo vệ khỏi truy cập trái phép và các vi phạm tiềm ẩn.

Thứ hai, chúng em đã kết hợp AES-GCM-256 để duy trì tính toàn vẹn của dữ liệu. Chế độ mã hóa này cung cấp khả năng xác thực và bảo mật, đảm bảo dữ liệu không bị giả mạo trong quá trình truyền hoặc lưu trữ. Ngoài ra, giải pháp của chúng em còn cung cấp tính năng băm tệp, cho phép người dùng kiểm tra tính toàn vẹn của tệp và phát hiện mọi sửa đổi trái phép.

Cuối cùng, chúng em đã triển khai cơ chế kiểm soát truy cập linh hoạt bằng cách sử dụng kỹ thuật Mã hóa dựa trên thuộc tính chính sách mã hóa (CP-ABE). Cách tiếp cận này cho phép chúng em có thể cung cấp khả năng kiểm soát truy cập chi tiết cho người dùng theo cách vừa linh hoạt vừa hiệu quả. Người dùng có thể được cấp quyền truy cập vào dữ liệu cụ thể dựa trên thuộc tính của họ, chẳng hạn như vai trò hoặc bộ phận của họ mà không cần phải quản lý riêng quyền truy cập cho từng người dùng.

Nhìn chung, giải pháp của chúng em cung cấp một hệ thống bảo mật toàn diện và hiệu quả nhằm giải quyết các vấn đề quan trọng về xác thực người dùng, tính toàn

vẹn dữ liệu và kiểm soát truy cập. Bằng cách sử dụng các kỹ thuật này, chúng em đã phát triển một giải pháp có thể cung cấp quyền truy cập an toàn vào dữ liệu nhạy cảm, đồng thời bảo vệ khỏi các mối đe dọa và vi phạm tiềm ẩn

5.2. Hạn chế

Song với kết quả đạt được, thì đồ án của chúng em vẫn còn một số tồn động chưa giải quyết được như:

- Dữ liệu mã hoá chưa thể truy vấn trực tiếp được mà phải thông qua việc tải về và giải mã.
- Việc sửa dữ liệu còn hạn chế, phức tạp, chưa được thuận tiện với người sử dụng.
- Giao diện sản phẩm còn khá đơn giản và chưa bắt mắt.
- Chưa phát triển được sản phẩm trên môi trường Windows.

5.3. Lời kết

- Từ những kết quả và hạn chế trên, trong quá trình thực hiện đồ án, chúng em đã học hỏi được rất nhiều điều bổ ích, hỗ trợ cho công việc trong tương lai sau này. Nhóm sẽ cố gắng phát triển và duy trì sản phẩm đồ án trên Github, cố gắng khắc phục được những hạn chế còn tồn động.

- Cuối cùng, một lần nữa chúng em xin chân thành cảm ơn thầy Nguyễn Ngọc Tự, thầy đã không ngại khó khăn, luôn giúp đỡ chúng em trong quá trình thực hiện đồ án và luôn nhiệt tình giảng dạy trong học tập. Chúng em xin gửi tặng thầy những lời chúc tốt đẹp nhất, chúc thầy luôn vui vẻ, hạnh phúc và thành công trong sứ mệnh cao cả của mình.

DANH MỤC TÀI LIỆU THAM KHẢO

1. Campagna, M., & Gueron, S. (2019). Key management systems at the cloud scale. *Cryptography*, 3(3), 23.
2. Mozaffari-Kermani, M., & Reyhani-Masoleh, A. (2011). Efficient and high-performance parallel hardware architectures for the AES-GCM. *IEEE Transactions on Computers*, 61(8), 1165-1178.
3. Odelu, V., Das, A. K., Rao, Y. S., Kumari, S., Khan, M. K., & Choo, K. K. R. (2017). Pairing-based CP-ABE with constant-size ciphertexts and secret keys for cloud environment. *Computer Standards & Interfaces*, 54, 3-9.
4. Agrawal, Shashank, and Melissa Chase. "FAME: Fast attribute-based message encryption." In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 665-682. 2017.
5. Ed-Daibouni, M., Lebbat, A., Tallal, S., & Medromi, H. (2016). Toward a new extension of the access control model ABAC for cloud computing. In *Advances in Ubiquitous Networking: Proceedings of the UNet'15 1* (pp. 79-89). Springer Singapore.