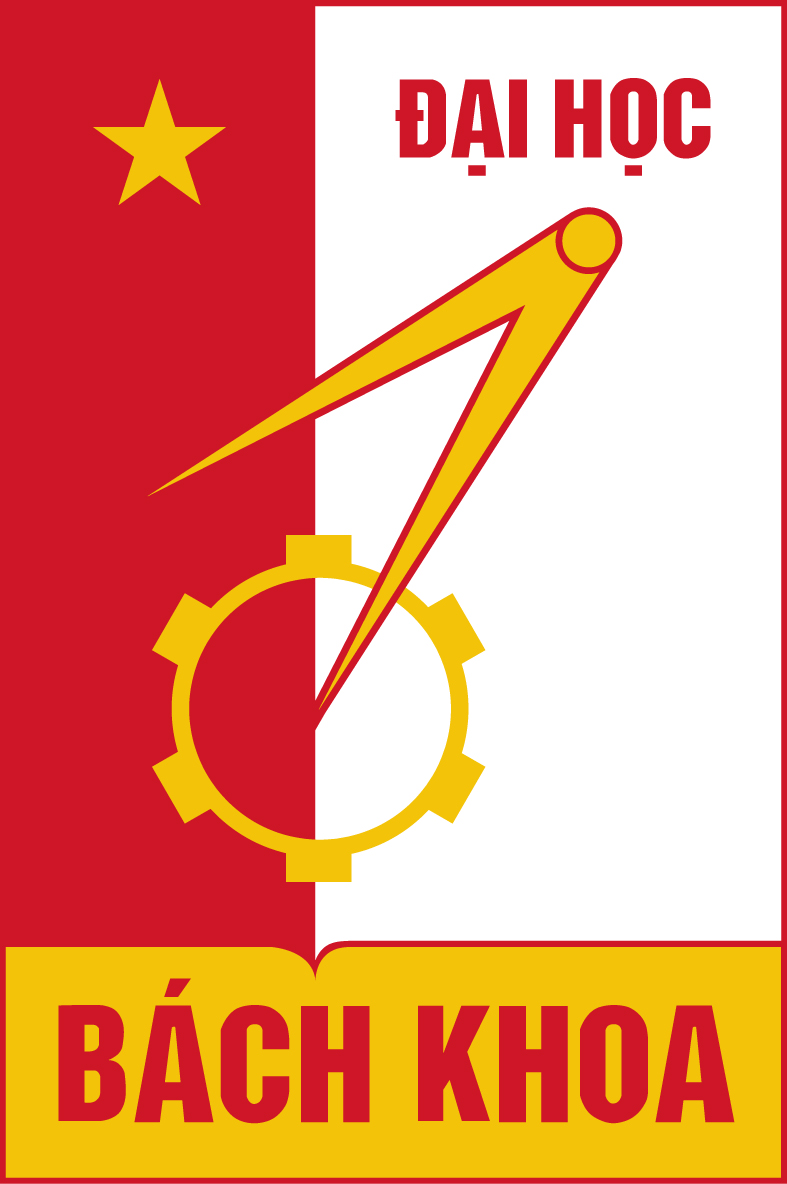
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

─────── \* ───────



Nhập môn An toàn Thông tin (IT4015)

**Đề tài: ỨNG DỤNG CHIA SẺ VỊ TRÍ**

Sinh viên thực hiện:

Mai Trường Sơn – 20183620

Đặng Quang Huy – 20183551

Đào Tùng Dương – 20183509

Giáo viên hướng dẫn: **TS. Trần Vĩnh Đức**

*Hà Nội, tháng 7 năm 2021*

Mục lục

[Danh mục hình ảnh 1](#_Toc76922441)

[1. Giới thiệu đề tài 2](#_Toc76922442)

[2. Cơ sở lý thuyết 3](#_Toc76922443)

[2.1. Phát biểu tổng quát bài toan 3](#_Toc76922444)

[2.2. Luồng thông tin trao đổi giữa Alice và Bob 3](#_Toc76922445)

[2.3. Tính đúng đắn và bảo mật của luồng thông tin 4](#_Toc76922446)

[2.4. Sinh cặp khoá chung () 4](#_Toc76922447)

[3. Thiết kế ứng dụng 5](#_Toc76922448)

[3.1. Database 5](#_Toc76922449)

[3.2. Biểu đồ usecase tổng quan 6](#_Toc76922450)

[3.3. Đặc tả usecase “Tìm quanh đây” 7](#_Toc76922451)

[4. Công nghệ sử dụng và mã nguồn 8](#_Toc76922452)

[5. Tổng kết 9](#_Toc76922453)

[6. Tài liệu tham khảo 10](#_Toc76922454)

# Danh mục hình ảnh

[Hình 1: Luồng thông tin trao đổi giữa Alice và Bob 3](#_Toc76922456)

[Hình 2: Cơ sở dữ liệu 5](#_Toc76922457)

[Hình 3: Biểu đồ use case tổng quan 6](#_Toc76922458)

[Hình 4: Biểu đồ hoạt động use case: Tìm quanh đây 7](#_Toc76922459)

# Giới thiệu đề tài

Trong thời đại bùng nổ về thông tin như hiện nay, việc chúng ta sẵn sàng chia sẻ thông tin định vị của cá nhân cho các ứng dụng, trang web hay một nhà cung cấp dịch vụ nào đó đang trở nên rất phổ biến. Nhìn về lợi ích, điều này giúp người dùng chúng ta thuận tiện hơn trong việc sử dụng các ứng dụng công nghệ vào đời sống (theo dõi thời tiết tại nơi mình ở, đặt xe, giao hàng, …).

Tuy nhiên, bên cạnh những điểm lợi đó, nhiều người vẫn tỏ ra khá cẩn trọng khi quyết định chia sẻ thông tin về vị trí của mình. Thực tế chỉ ra có khá nhiều ứng dụng theo dõi thông tin định vị của người dùng vào những mục đích xấu (theo dõi, ăn cắp thông tin, …)

Chính vì vậy, chúng em đề xuất việc xây dựng ra một ứng dụng cho phép người dùng chia sẻ vị trí của mình với các đặc điểm bảo mật được mô tả trong bài toán cụ thể sau:

* Alice đang đi nghỉ mát tại Nha Trang, Alice muốn biết xem người bạn thân Bob của mình liệu có đang ở Nha Trang hay không, nếu Bob cũng ở Nha Trang thì họ có thể hẹn gặp nhau để ăn một bữa tối.
* Tuy nhiên, nếu Bob không ở Nha Trang, thì Alice lại không muốn tiết lộ cho Bob rằng mình đang ở Nha Trang. Ở chiều ngược lại, Bob cũng mong muốn điều tương tự, tức là không tiết lộ cho Alice vị trí nếu không ở gần Alice.

# Cơ sở lý thuyết

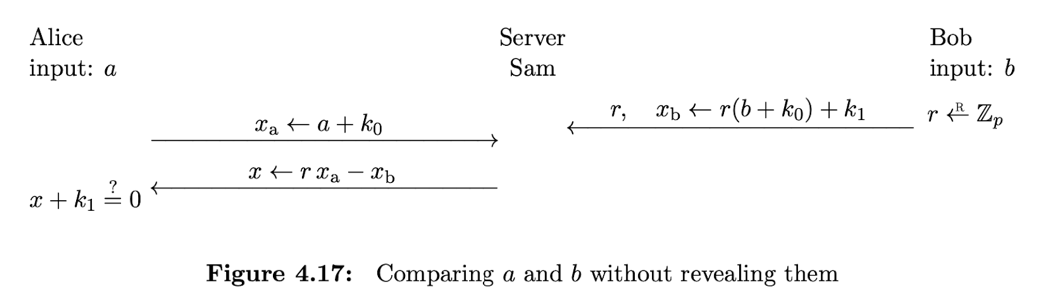
## Phát biểu tổng quát bài toan

Qua bài toán đặt ra ở trên, ta có thể phát biểu một cách tổng quát như sau:

* Ứng dụng cho phép người dùng chia sẻ vị trí của mình, kết quả trả về sẽ là những người thân đang ở “gần” (tạm coi “gần” ở đây là chung một tỉnh/ thành phố).
* Những người không ở “gần” sẽ không hiển thị thông tin vị trí với người dùng và ngược lại, thông tin của người dùng sẽ không hiển thị với những người ở xa.

Với mô tả như trên, ta thấy rằng không thể để người dùng (ví dụ là Alice và Bob) giao tiếp trực tiếp với nhau. Một giải pháp đơn giản được đưa ra là có một người bạn chung Sam, Sam sẽ liên hệ cho Alice và Bob để hỏi thông tin, sau đó cung cấp kết quả cho Alice. Tuy nhiên không thể chắc chắn rằng Sam không tiết lộ thông tin vị trí của Bob cho Alice và ngược lại. Vậy nên bên cạnh sử dụng một trung gian, ta cũng sẽ đảm bảo trung gian này không thể biết được chính xác thông tin vị trí của người dùng.

## Luồng thông tin trao đổi giữa Alice và Bob



Hình 1: Luồng thông tin trao đổi giữa Alice và Bob

Thông tin vị trí của Alice và Bob lần lượt là và . Hai người cũng đồng thời chia sẻ với nhau một cặp khoá chung **()** mà Sam không được biết.

1. Bob sinh ra một số ngẫu nhiên, đồng thời gửi và đến cho Sam, với được tính bởi công thức:
2. Khi muốn biết vị trí của Bob, Alice sẽ gửi thông tin cho Sam.
3. Sam nhận được thông tin từ Alice và Bob, Sam sẽ trả về cho Alice thông tin
4. Alice sẽ tính giá trị của biểu thức . Nếu thì Alice và Bob đang ở gần nhau, trái lại thì Alice và Bob đang ở xa nhau.

## Tính đúng đắn và bảo mật của luồng thông tin

* Sam nhận được thông tin mà không biết cặp khoá **()** nên không thể biết được vị trí của Alice và Bob.
* Alice nhận được thông tin từ Sam nên cũng không thể suy ra được vị trí của Bob.
* Ở đây vai trò của Alice và Bob là không tương đương, Alice chỉ khi cần biết vị trí mới chia sẻ thông tin cho Sam, còn Bob mặc định sẽ định kỳ gửi thông tin cho Sam.
* Biểu thức Alice cần kiểm tra:
  + Nếu hay Alice ở gần Bob, giá trị
  + Nếu , Alice sẽ nhận được một giá trị ngẫu nhiên khác 0, phụ thuộc vào giá trị của . Chính vì vậy mà không thể suy ra được vị trí của Bob

## Sinh cặp khoá chung ()

Một vấn đề đặt ra trong luồng trao đổi thông tin giữa Alice và Bob, là việc sử dụng khoá chung **()**. Nếu cặp khoá này được sử dụng lại nhiều lần, việc bảo mật thông tin có thể không được đảm bảo. Ví dụ Alice 2 lần gửi thông tin vị trí và cho Sam với cùng cặp khoá **()** dưới dạng: và . Từ thông tin này, Sam có thể tính ra được giá trị .

Do vậy, giải pháp đưa ra là sử dụng một hàm giả ngẫu nhiên (PRF) để sinh cặp khoá cho mỗi lần trao đổi thông tin của Alice và Bob. Giả sử Alice và Bob có chung với nhau một **khoá cố định** . Ở phía Bob sẽ giữ một bộ đếm . Mỗi lần gửi thông tin mã hoá cho Sam, Bob sẽ tăng giá trị của bộ đếm lên 1; sau đó sinh ra cặp khoá mới như sau:

**(**

Tiếp đó, Bob gửi cho Sam thông tin gồm (**)**. Trước khi Alice gửi thông tin của mình cho Sam, Alice sẽ yêu cầu Sam gửi cho mình giá trị của bộ đếm gần nhất mà Bob gửi. Từ giá trị nhận được, Alice sẽ tính được ra cặp khoá mà Bob đã dùng cho phiên trao đổi này. Sau đó thông tin được trao đổi như luồng phía trên.

Lưu ý, khi nhận được giá trị mớitừ Sam, Alice phải kiểm tra rằng giá trị này phải lớn hơn giá trị lần gần nhất; tránh trường hợp Sam có ý đồ xấu, cố tình gửi lại giá trị cũ của bộ đếm, khiến khoá của Alice sinh ra khác với khoá mà Bob sử dụng.

# Thiết kế ứng dụng

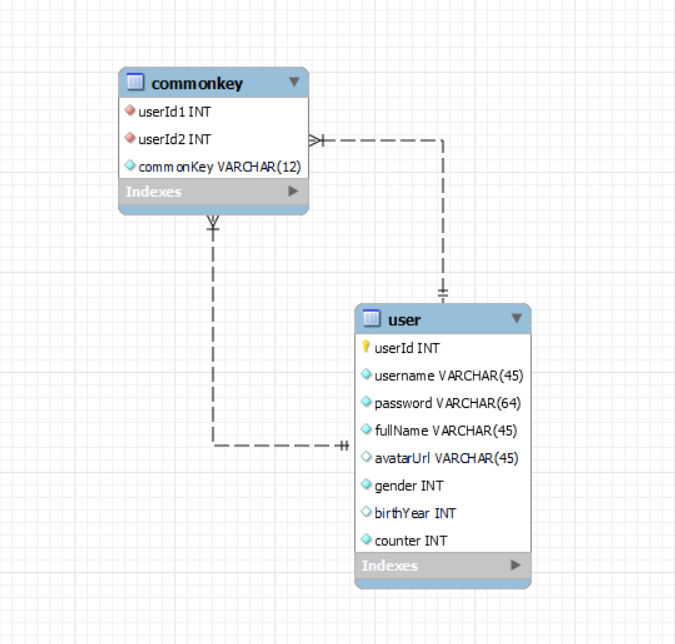
Với những phân tích phía trên, ta coi chương trình đóng vai trò như người bạn chung Sam. Người dùng trong chương trình của nhóm sẽ cùng lúc đóng 2 vai trò: Alice và Bob. Cụ thể, khi sử dụng ứng dụng, mỗi người dùng đều phải chấp nhận gửi thông tin dạng cho hệ thống để đảm bảo hoạt động chung (như Bob); còn khi người dùng có nhu cầu tìm kiếm những người ở gần (như Alice), họ sẽ gửi thông tin dạng cho hệ thống. Khi đó kết quả danh sách người ở gần sẽ được trả về cho người dùng.

Với hệ thống gồm người dùng, mỗi người sẽ mang một bộ đếm (vì đều mang vai trò của Bob). Mỗi cặp người dùng sẽ chia sẻ một khoá cố định chung, nên tổng sẽ có khoá chung.

Để đơn giản hoá cho việc xây dựng ứng dụng, nhóm quyết định sẽ lưu trữ các dữ liệu về bộ đếm và khoá chung vào cơ sở dữ liệu (tương đương với việc Sam biết khoá chung của mọi cặp người dùng). Điều này không ảnh hưởng đến tính riêng tư của người dùng do server không lưu trữ khoá cho PRF() sinh khoá phiên.

Mô hình xây dựng dưới dạng Client – Server. Cụ thể server hệ thống đóng vai trò như nhận yêu cầu từ client, gửi thông tin khoá chung và bộ đếm cho những client liên quan, nhận thông tin dạng từ phía client. Về phía client, nhận khoá chung và bộ đếm từ server, sinh khoá phiên, mã hoá thông tin dạng gửi về cho server. Luồng hoạt động cụ thể sẽ được mô tả rõ hơn trong phần đặc tả use case.

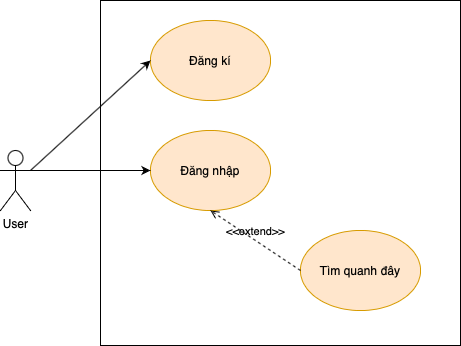
## Database



Hình 2: Cơ sở dữ liệu

* Bảng User:
  + userid: id của người dùng
  + username: Tên đăng nhập
  + password: Mật khẩu đã được mã hóa bằng SHA256
  + fullName: Họ và tên
  + avatarUrl: Đường dẫn đến file avatar
  + gender: Giới tính
  + birthYear: Năm sinh
  + counter: Bộ đếm
* Bảng CommonKey:
  + userid1, userid2: id của người dùng, khóa ngoại của bảng
  + commonKey: khóa chung k của cặp người dùng có id là userid1 và userid2

## Biểu đồ usecase tổng quan



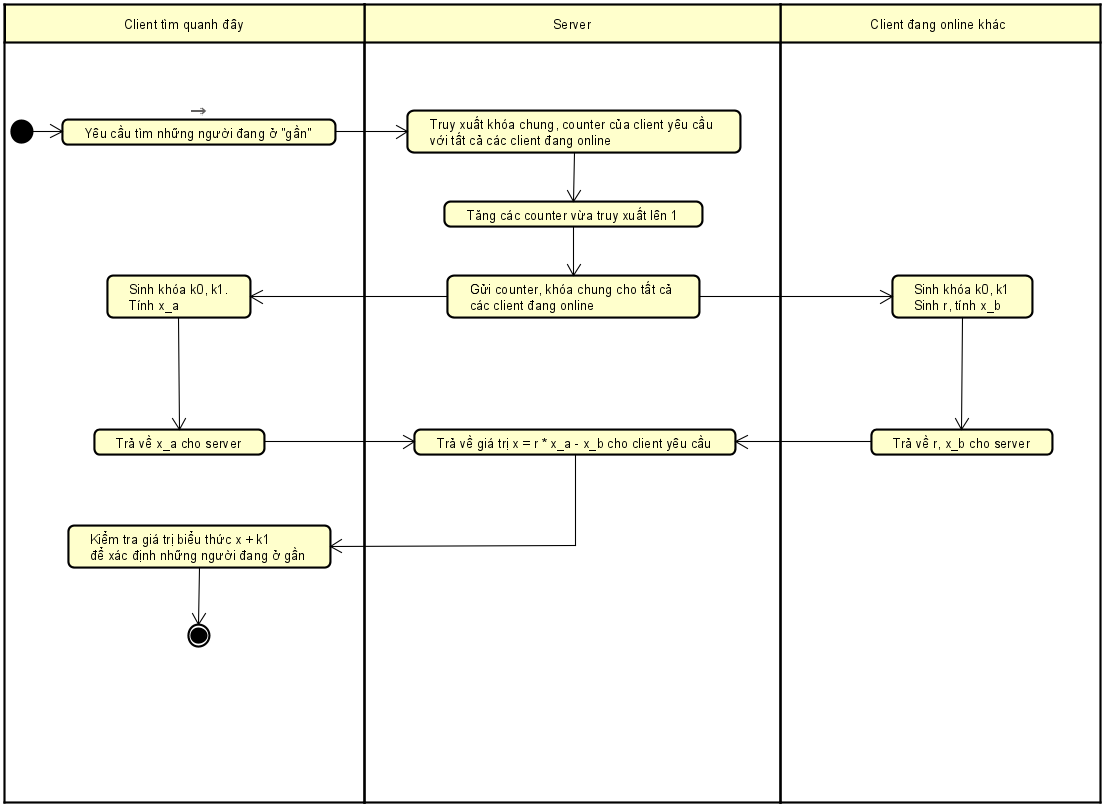
Hình 3: Biểu đồ use case tổng quan

Sau khi đăng ký, hệ thống sẽ tự động sinh khóa chung với tất cả các user còn lại trong hệ thống, đồng thời counter sẽ được khởi tạo bằng 0.

Khi đăng nhập, hệ thống sẽ auto locate địa điểm của người dùng. Đồng thời, hệ thống ghi nhận người dùng đang online, chuẩn bị cho use case “Tìm quanh đây”.

Use case “Tìm quanh đây” sẽ trả về những người đang sử dụng hệ thống (online) mà ở “gần” với client chọn sử dụng chức năng này.

## Đặc tả usecase “Tìm quanh đây”



Hình 4: Biểu đồ hoạt động use case: Tìm quanh đây

Khi một người dùng yêu cầu “Tìm quanh đây”:

* Hệ thống sẽ truy xuất trong database và trả về counter mới, khóa chung của các người dùng đang online.
* Sau đó, trên máy client của mỗi người dùng online sẽ sinh khóa **(** dựa vào khóa chung và counter. **(** lần lượt là 8 byte đầu và 8 byte cuối của mã hóa 3-DES với các khoá lần lượt là: ‘12345678’, ‘abcdefgh’ và khóa chung. Giá trị được ngẫu nhiên trong khoảng (10000, 4000000000).
* Thông tin vị trí của người dùng sẽ được auto locate bằng API hoặc được người dùng tự nhập; sau đó sẽ được mã hóa thành các số nguyên từ 1 đến 63, mỗi số ứng với 1 vị trí . Các giá trị , được tính bởi công thức: , .
* Người dùng “tìm quanh đây” sẽ gửi lại giá trị cho server. Còn với những user đang online còn lại sẽ gửi lại giá trị , cho server.
* Với mỗi bộ giá trị **(** nhận được, server sẽ trả lại giá trị cho người dùng yêu cầu “Tìm quanh đây”.
* Dựa vào việc kiểm tra giá trị biểu thức , người dùng sẽ biết được có ai đang ở “gần” mình hiện tại.

# Công nghệ sử dụng và mã nguồn

* 1. Công nghệ sử dụng:
* Ngôn ngữ: python 3.x
* Thư viện: mã hoá pycryptodome, Flask (flask, flask-restful), mysql-connector-python
* GUI: pyQt5
* CSDL: MySQL (lưu khoá chung và bộ đếm )
  1. Mã nguồn:
* Server: [https://github.com/maitruongson-vn107/locationshare-server-v2](https://www.google.com/url?q=https://github.com/maitruongson-vn107/locationshare-server-v2&sa=D&source=editors&ust=1625621084752000&usg=AOvVaw1db1QOIHTug188aJzD8KpK)
* Client: [https://github.com/maitruongson-vn107/locationshare-client-v2](https://www.google.com/url?q=https://github.com/maitruongson-vn107/locationshare-client-v2&sa=D&source=editors&ust=1625621084752000&usg=AOvVaw1pIUaA7YX0T5p7NC7C0ucC)
  1. Demo chương trình

Link Youtube: <https://youtu.be/l0xuxoGyahI>

1. Tổng kết

Chương trình xây dựng đảm bảo các yêu cầu bảo mật đặt ra, thực hiện đúng luồng trao đổi thông tin mật mã theo lí thuyết. Các thao tác sử dụng dễ dàng, thuận tiện; giao diện đơn giản, thân thiện với người dùng.

Tuy nhiên, bên cạnh những điểm đã thực hiện được, chương trình còn nhiều điểm phải cải thiện như giao diện chưa bắt mắt, việc lấy vị trí người dùng phụ thuộc vào API của bên thứ 3 hay thông tin khoá vẫn phải lưu trên database chung. Ngoài ra cũng có nhiều chức năng có thể được phát triển thêm như: tạo nhóm với những người ở gần; kết bạn, theo dõi; nhắn tin với những người ở gần,… Nếu có thời gian tiếp tục phát triển chương trình, nhóm sẽ khắc phục những tồn tại của chương trình cũng như bổ sung những chức năng mới cho chương trình.

Quá trình thực hiện bài tập lớn cũng đã đem lại cho các thành viên trong nhóm cơ hội ôn tập, rèn luyện kiến thức của học phần Nhập môn An toàn thông tin cũng như kĩ năng lập trình xây dựng sản phẩm. Chúng em rất mong nhận được thêm lời góp ý nhận xét từ thầy cũng như các bạn trong lớp để nhóm có thể hoàn thiện sản phẩm hơn trong tương lai.

Nhóm chúng em xin chân thành cảm ơn!

1. Tài liệu tham khảo

* Dan Boneh and Victor Shoup, *A Graduate Course in Applied Cryptography,* Jan 2020, p. 163-165