VIETNAM NATIONAL UNIVERSITY – HO CHI MINH CITY

UNIVERSITY OF SCIENCE

# REPORT: SEMINAR 3rd

# TOPIC: END-TO-END ENCRYPTION

## Course: Applied Cryptography

| | | |
|---|---|---|
| **TEACHER:** | Truong Toan Thinh | |
| | Tran Minh Triet | |
| **ASSISTANT TEACHING:** | Luong Vi Minh | |
| **STUDENTS:** | Nguyen Nhat Quan | 20127066 |
| | Hoang Huu Minh An | 20127102 |
| | Tran Anh Huy | 20127192 |
| | Truong Gia Thinh | 20127338 |

Ho Chi Minh, 16-04-2023.

# Mục Lục

## I. Thông tin nhóm:

Group 6:

| STT | MSSV | Họ và Tên |
|---|---|---|
| 1 | 20127066 | Nguyễn Nhật Quân |
| 2 | 20127102 | Hoàng Hữu Minh An |
| 3 | 20127192 | Trần Anh Huy |
| 4 | 20127338 | Trương Gia Thịnh |

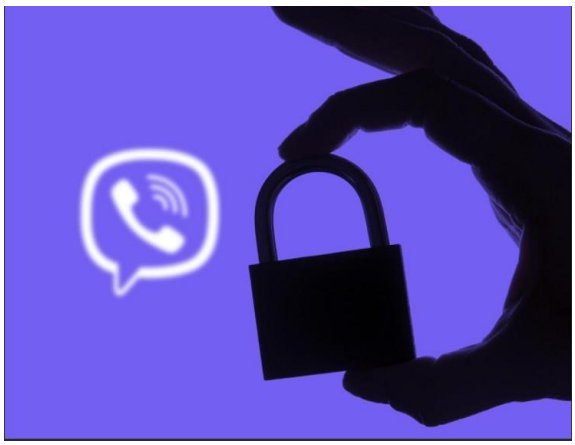
## II. Giới thiệu về chủ đề mới:

Xu thế hiện nay, Sự bùng nổ công nghệ trong những năm gần đây. Người dùng cũng cần đòi hỏi về an toàn riêng tư của mình khi truyền tải thông tin trên không gian mạng. Để đáp ứng được nhu cầu này, việc mã hóa đầu cuối là một phương pháp mã hóa dữ liệu để bảo vệ riêng tư cho người dùng hiệu quả.

Người dùng thích sử dụng mã hóa đầu cuối là tính riêng tư và bảo mật mà nó mang lại. Khi sử dụng mã hóa đầu cuối, thông tin người dùng được mã hóa trên thiết bị và chỉ có người dùng cuối cùng mới có thể giải mã nó. Điều này đảm bảo bên thứ 3 không thể truy cập hoặc xem thông tin truyền đi, bao gồm các tổ chức, nhà cung cấp dịch vụ hoặc kẻ tấn công trên mạng.

Mục đích có thể trải nghiệm và học hỏi tìm hiểu về một cơ chế bảo mật phổ biến hiện nay, nhóm chúng em đã bàn bạc và đi đến quyết định sẽ thực hiện nghiên cứu về chủ đề: Mã hóa đầu cuối:

- Tính áp dụng thực tế cao: Đây là cơ chế mã hóa cần phải có khi muốn xét về độ an toàn và bảo mật của một ứng dụng nhắn tin trực truyến.
- Tính bảo mật: Mã Hóa Đầu Cuối bảo mật hoàn toàn tin nhắn người dùng với bên thứ 3 kể cả server chính

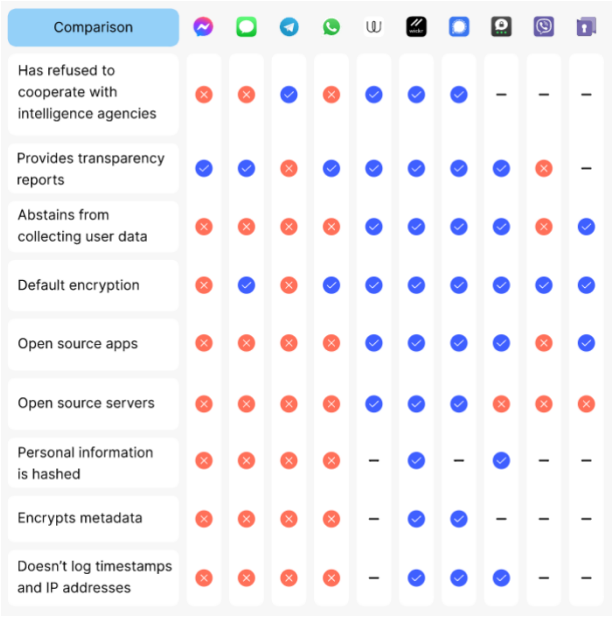


## III. Khảo sát thực tế:

Hiện nay, mã hóa đầu cuối đang trở nên thành các xu hướng phổ biến trong các ứng dụng tin nhắn và gọi video trực tuyến. Các ứng dụng như Signal, WhatsApp và Telegram,… đều sử dụng mã hóa tin nhắn bảo vệ sự riêng tư cho người dùng.

| STT | App | | E2EE | chú thích | Phương pháp mã hóa |
|---|---|---|---|---|---|
| 1 | Messenger | | E2EE | | |
| 2 | Zalo | | E2EE | | |
| 3 | Discord | | No | | Transport Layer Security |
| 4 | WhatApps | | E2EE | | |
| 5 | Telegram | | E2EE | | |
| 6 | Wechat | | No | | symmetric encryption<br>WeChat uses a hybrid RSA-AES encryption scheme for asymmetric encryption |
| 7 | Wire | | E2EE | | |
| 8 | Skype | | No | | Skype uses the Advanced Encryption Standard (AES) algorithm for symmetric encryption and the RSA algorithm for asymmetric encryption |
| 9 | Line | | E2EE | | |
| 10 | KakaoTalk | | E2EE | | |
| 11 | Snapchat | | E2EE | | |
| 12 | Google Hangouts | | No | Google Hangouts doesn't have an end to end encryption. Google describes the Hangouts encryption as functional, as it encrypts messages in transit, i.e., when they get sent. | Google Hangouts uses a combination of Secure Real-time Transport Protocol (SRTP) and Transport Layer Security (TLS) |
| 13 | Viber | | E2EE | | |
| 14 | Slack | | No | | Data is encrypted using the Advanced Encryption Standard (AES) algorithm before being stored, and the encryption keys are managed using the Key Management Service (KMS) provided by Amazon Web Services. |
| 15 | Lark | | E2EE | | |
| 16 | Tango | | E2EE | | |
| 17 | iMessage | | E2EE | | |
| 18 | Signal | | E2EE | Most secure messaging app | |
| 19 | Zoom | | E2EE | | |
| 20 | Omegle | | No | | Users must use VPN to hide IP, the message is not encypted |

*HÌNH 1. SO SÁNH CÁCH THỨC MÃ HÓA 1*



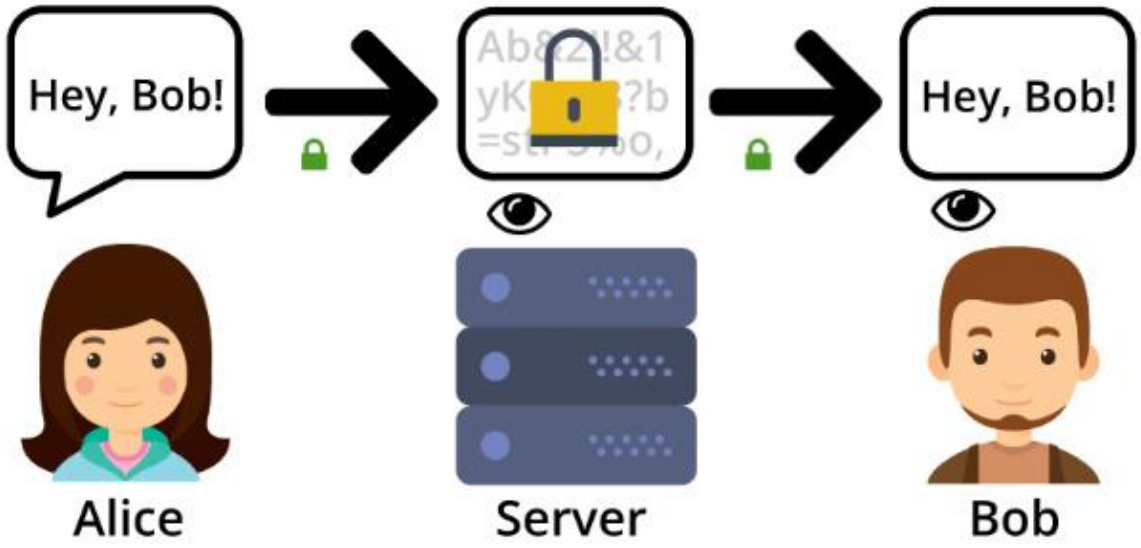

*HÌNH 2. SO SÁNH CÁC ỨNG DỤNG NHẮN TN 1*

Tuy nhiên, việc sử dụng mã hóa đầu cuối làm cho nỗ lực của chính phủ và các cơ quan chức năng nhằm chống lại tổ chức tội phạm lừa đảo, khủng bố, khiêu dâm trẻ em về mặt kĩ thuật là không thể.

## IV. Mã hóa đầu cuối là gì:

Bạn mã hóa tin nhắn/hình ảnh cần gửi và nó chuyển qua Internet dưới dạng mã 'bí mật'. Sau đó chỉ người nhận mới có thể giải mã mã 'bí mật' này.

=> Quá trình này được gọi là mã hóa đầu cuối.




## V. Các hướng tiếp cận:

Một số hướng tiếp cận của mã hóa đầu cuối:

- Giao thức mã hóa đầu cuối như là Signal Protocol, OMEMO hay Axolotl Protocol để đảm bảo riêng tư.
- Kỹ thuật phân tán khóa công khai (decentralized public key infrastructure - DPki):để đảm bảo tính báo mật tốt hơn
- Mã hóa đa phương tiện để không chỉ bảo vệ mà cả âm thanh, hình ảnh, video...
- Mã hóa đa nhân tố; có thể kết hợp mã hóa với xác thực bằng vân tay, khuôn mặt, hoặc mã OTP (one-time-password).
- Quantum Key Distribution (QKD): QKD kỹ thuật sử dụng các tính chất đặc biệt của vật lý lượng tử để tạo ra các khóa bí mật.
- Functional Encryption: Functional Encryption là một kỹ thuật người dùng chia sẻ dữ liệu với những người khác mà chỉ cho phép họ truy cập vào những phần cụ thể.

- ZKP (Zero-knowledge Proof): trong mã hóa đầu cuối là phương pháp xác thực thông tin giữa 2 bên mà không cần không cần tiết lộ bất kì thông tin bí mật nào ban đầu.
- Mã hóa đa khóa (multi-key encryption): Mã hóa đa khóa là một hướng tiếp cận mà mỗi người dùng có nhiều khóa mã hóa khác nhau để bảo vệ thông tin. Với hướng tiếp cận này, tin nhắn được mã hóa bởi nhiều khóa khác nhau, mỗi khóa chỉ có thể được giải mã bởi một người dùng cụ thể.
- Mã hóa đám mây: Mã hóa đầu cuối trong môi trường đám mây là một hướng tiếp cận mới đang được phát triển để đảm bảo tính bảo mật và sự riêng tư của dữ liệu khi được lưu trữ trên các dịch vụ đám mây. Với hướng tiếp cận này, thông tin sẽ được mã hóa trên thiết bị của người dùng trước khi được tải lên đám mây, đảm bảo tính bảo mật và sự riêng tư của dữ liệu.
- Mã hóa dựa trên BlockChain.

## VI. Phương thức:

Có nhiều phương thức được sử dụng để mã hóa và giải mã thông tin:

- Symetric-key cryptography: sử dụng 1 khóa để giải mã các thông điệp như là AES, DES, …
- Public-key cryptography: phương thức này sử dụng hai khóa khác nhau để mã hóa và giải mã thông điệp. Khóa công khai (public key) được sử dụng để mã hóa thông điệp, trong khi khóa bí mật (private key) được sử dụng để giải mã thông điệp. Các thuật toán phổ biến trong public-key cryptography bao gồm RSA, ECC, và Diffie-Hellman.
- Hash functions: phương thức này được sử dụng để tạo ra một bản tóm tắt (digest) đại diện cho một thông điệp. Bản tóm tắt này có độ dài cố định và không thể được giải mã. Hash functions được sử dụng để xác thực tính toàn vẹn của thông điệp và để tạo ra chữ ký số (digital signature).

Ngoài ra còn nhiều phương thức có thể áp dụng

## VII. Ưu điểm và nhược điểm:

- **Ưu điểm:**
  - Tính bảo mật cao: Thông điệp được mã hóa tại điểm bắt đầu và được giải mã tại điểm cuối
  - Độ tin cậy cao: Các máy chủ trung gian không có khả năng truy cập và đọc thông điệp bảo mật
  - Kiểm soát truy cập: Người sử dụng hoàn toàn kiểm soát nội dung của thông điệp và ai có thể xem nó
- **Nhược điểm:**
  - Siêu dữ liệu như ngày, giờ và tên người tham gia không được mã hóa.

- Không có khả năng giám sát thông điệp: chính phủ không thể giám sát hoặc kiểm tra nội dung của thông điệp. Điều này có thể làm giảm khả năng phát hiện các hoạt động bất hợp pháp hoặc nguy hiểm.
- Trong một số trường hợp, có thể có xảy ra cuộc tấn công Man-in-the-Middle

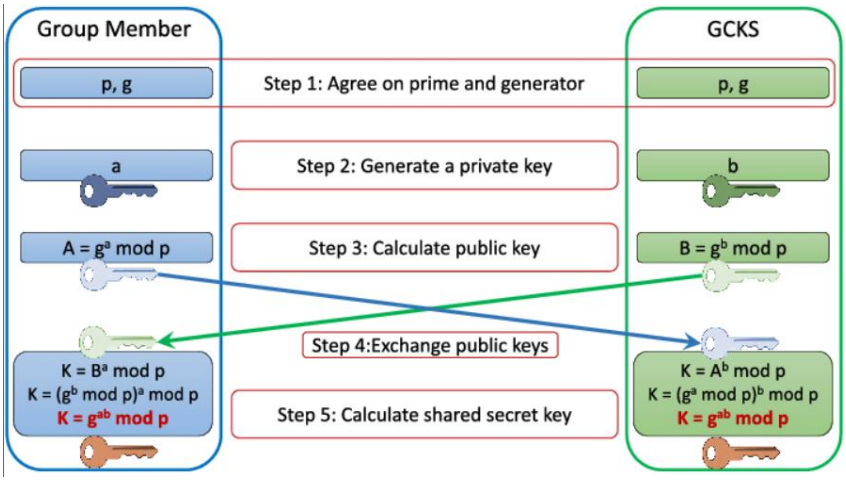
## VIII. Ý tưởng tiếp cận:

Việc ứng dụng mã hóa đầu cuối vào việc trao đổi tin nhắn có rất nhiều và đòi lượng kiến thức liên quan, nên nhóm đang cân nhắc các hướng tiếp cận đơn giản và dễ triển khai.

- Mã hóa tin nhắn bằng giải thuật: Diffie-Hellman, AES
- Kênh truyền thông tin: TCP/IP, SSL/TLS, SSH
- Hướng tiếp cận: ZKP (Zero-knowledge Proof) để xác thực thông tin tính toàn vẹn và xác thực của thông điệp mà không tiết lộ thông tin nào cả hoặc là chứ ký số hoặc ký khóa.

## IX. Thuật toán Diffie-Hellman:

1) **Định nghĩa**: Diffie-Hellman là một thuật toán mã học được sử dụng để thiết lập khóa chung (shared secrect) giữa hai bên trong quá trình truyền tin an toàn
2) **Lịch sử phát triển:** Thuật toán này được phát minh bởi Whitfield Diffie và Martin Hellman vào năm 1976
3) **Cơ sở toán học:** Thuật toán dựa trên giả định tính toán logarithmic rời rạc trên một số nguyên tố lớn rất khó khăn. Giả định số nguyên tố cho việc tìm ra các thừa số nguyên tố của một số nguyên tố lớn là khó khắn.
4) **Mô tả thuật toán:**




5) **Độ an toàn:** Độ an toàn của Diffie-Hellman là khá cao. Tuy nhiên Một trong những vấn đề của Diffie-Hellman là khả năng tấn công Man-in-the-Middle. Nếu một kẻ tấn công đang giữa hai bên trao đổi khóa, họ có thể thay đổi khóa để cả hai bên không phát hiện ra điều này và tiếp tục trao đổi thông tin. Để khắc phục vấn đề này, các hệ thống sử dụng Diffie-Hellman thường kết hợp

với các phương pháp xác thực để đảm bảo tính toàn vẹn của thông tin.Một vấn đề khác của Diffie-Hellman là khả năng tấn công bằng cách sử dụng brute force để tìm kiếm khóa. Tuy nhiên, vấn đề này có thể được giải quyết bằng cách sử dụng các tham số an toàn với độ dài đủ lớn và cập nhật thường xuyên để tránh các cuộc tấn công brute force.

6) **Nhận xét:**

Lợi thế khi sử dụng thuật toán Diffie-Hellman là:

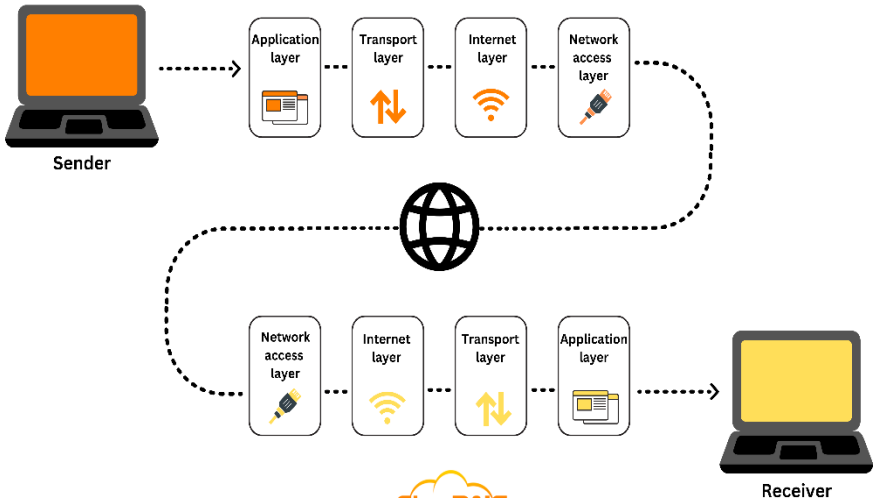
- Cung cấp tính bảo mật cao cho việc trao đổi khóa mật, đặc biệt là khi sử dụng kết hợp với các thuật toán mã hóa mạnh như AES hay RSA.
- Thuật toán Diffie-Hellman được sử dụng rộng rãi trong các ứng dụng mã hóa đầu cuối, đảm bảo tính riêng tư cho các giao dịch trực tuyến, chẳng hạn như các cuộc gọi thoại hay trao đổi tin nhắn giữa các thiết bị.
- Không yêu cầu việc trao đổi khóa mật qua kênh an toàn, giúp giảm tải cho hệ thống mạng và tăng tốc độ trao đổi khóa.

Tuy nhiên, việc sử dụng Diffie-Hellman cũng có một số bất lợi:

- Thuật toán không cung cấp tính xác thực cho các bên tham gia trao đổi khóa, do đó có thể bị tấn công giả mạo người dùng.
- Khi sử dụng Diffie-Hellman, một số kẽ hở bảo mật có thể được tận dụng để tấn công, đặc biệt là với các phiên bản cũ hơn của thuật toán. Các tấn công phổ biến bao gồm tấn công Man-in-the-Middle và các cuộc tấn công DDos.
- Các phương pháp để tăng cường tính an toàn của thuật toán, chẳng hạn như sử dụng khóa dài hơn hoặc kết hợp với các thuật toán khác, có thể ảnh hưởng đến tốc độ và hiệu suất của hệ thống.

## X. Kênh truyền thông tin TCP/IP:

1) **Kênh truyền TCP/IP là gì:** TCP/IP là một mô hình khái niệm (conceptual model) và một tập hợp các giao thức truyền thông dùng trong mạng Internet và các hệ thống mạng máy tính tương tự.
2) **Cách thức truyền dữ liệu:** Khi đã hình thành liên kết TCP, hai tiến trình tham gia liên kết có thể truyền dữ liệu cho nhau. Khi một tiến trình chuyển một chuỗi byte qua socket tới tầng giao vận, dữ liệu đó sẽ hoàn toàn do TCP trên máy đó quản lý và chịu trách nhiệm.TCP có thể coi như một hệ thống con của hệ điều hành chịu trách nhiệm nhận dữ liệu từ ứng dụng cục bộ cũng như nhận dữ liệu từ máy ở xa.



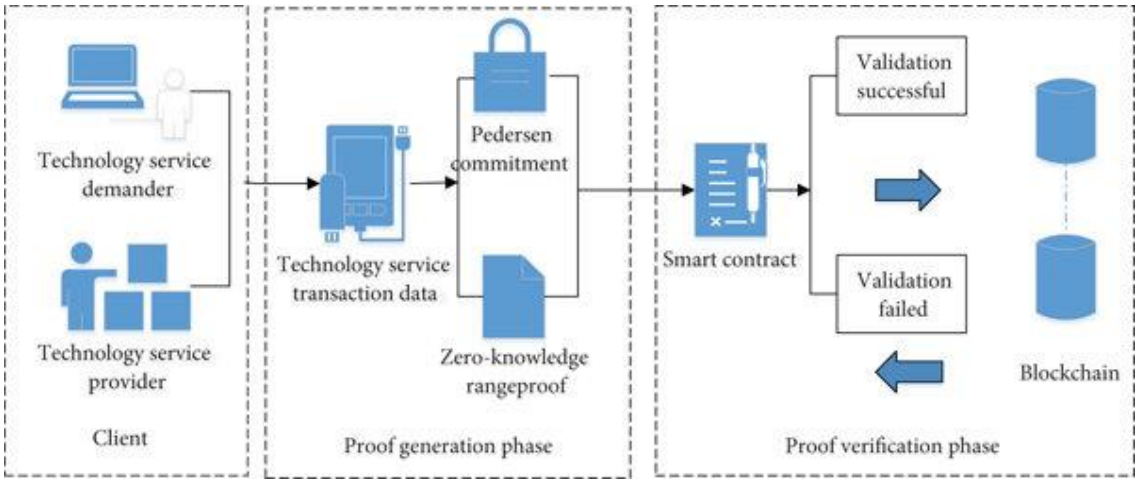

3) **Độ an toàn:** Việc sử dụng TCP/IP không đảm bảo tính an toàn cho các thông tin được truyền tải, vì giao thức này không cung cấp bất kỳ cơ chế bảo mật nào. Các thông tin truyền tải qua mạng bằng TCP/IP có thể bị đánh cắp, thay đổi hoặc giả mạo bởi các kẻ tấn công.
   ➔ Để giải quyết vấn đề này, các phương pháp mã hóa và xác thực như SSL/TLS và VPN được sử dụng để bảo vệ thông tin truyền tải qua mạng

## XI. Zero-knowledge Proof:

1) ***Định nghĩa:*** là một loại chứng thực tài liệu được sử dụng trong bảo mật thông tin. Nó cho phép một bên chứng minh cho bên còn lại rằng mình có một thông tin mà không cần phải tiết lộ thông tin đó cho bên còn lại. Điều này có nghĩa là bên còn lại không thể biết được thông tin đó mà chỉ biết rằng bên chứng minh thật sự có thông tin đó.
2) ***Một số biến thể:***
   - Non-Interactive Zero-Knowledge Proof (NIZKP): Trong NIZKP, bằng cách sử dụng một kỹ thuật được gọi là pre-processing, bên chứng minh có thể tạo ra một chứng minh mà không cần liên lạc trực tiếp với bên xác nhận.
   - Interactive Zero-Knowledge Proof (IZKP): Trong IZKP, bên chứng minh và bên xác nhận cần phải trao đổi thông tin với nhau. Các bước trao đổi thông tin này có thể được lặp đi lặp lại nhiều lần để tăng tính bảo mật.
   - Non-Black-Box Zero-Knowledge Proof (NBBZKP): Trong NBBZKP, bên chứng minh được phép sử dụng các phương thức không thuộc black-box để tạo ra chứng minh.
   - Statistical Zero-Knowledge Proof (SZKP): Trong SZKP, bên xác nhận chỉ cần tin rằng bằng cách sử dụng đúng cách, bên chứng minh sẽ tạo ra một chứng minh đúng với xác suất cao.
   - Quantum Zero-Knowledge Proof (QZKP): Trong QZKP, bên chứng minh và bên xác nhận sử dụng các thuật toán lượng tử để trao đổi thông tin.

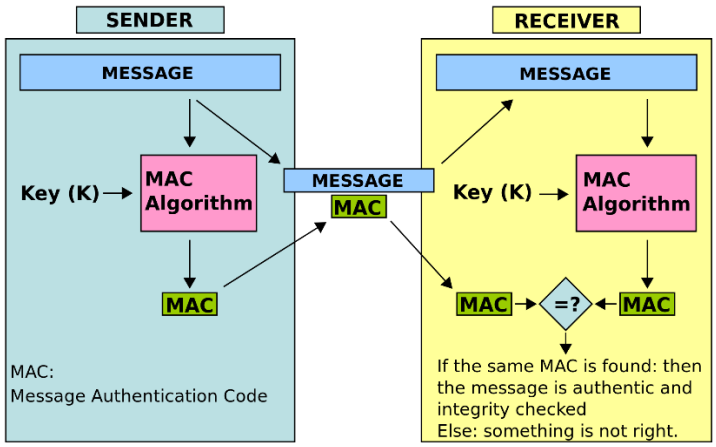
QZKP có tính bảo mật cao hơn so với các biến thể ZKP khác, nhưng hiện vẫn đang được nghiên cứu và phát triển.

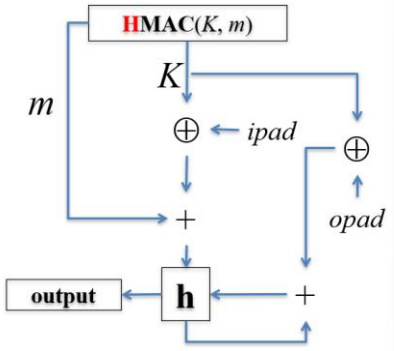
3) **Ứng dụng:** ZKP được sử dụng trong nhiều lĩnh vực, chẳng hạn như xác thực danh tính, chứng thực giao dịch tài chính, chứng thực thông tin nhạy cảm và các ứng dụng trong blockchain.ZKP được thiết kế để đảm bảo tính riêng tư và bảo mật cho các thông tin nhạy cảm, đồng thời giúp giảm thiểu sự phụ thuộc vào các bên trung gian trong quá trình chứng thực.
4) **Cách thức hoạt đông:**




## XII. Hàm HMAC:

1) **Định nghĩa:** là một mã xác thực thông điệp cho việc sử dụng một khóa mật mã dựa trên sức mạnh của hàm băm và khó bị collision attack thành công vì có khóa bí mật
2) **Phiên bản:** HMAC có thể được sử dụng với nhiều thuật toán hàm băm phổ biến, bao gồm MD5, SHA-1, SHA-256, và SHA-512. Tuy nhiên, các thuật toán hàm băm này đã bị tấn công và có các lỗ hổng bảo mật, do đó, các thuật toán hàm băm mới như SHA-3 được khuyến nghị để sử dụng trong HMAC.
3) **Cách thực hoạt động:**



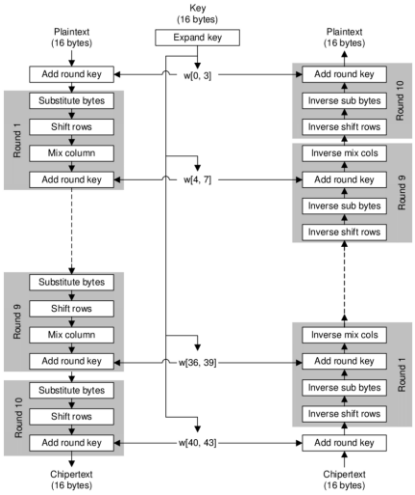



***Mã giả:***

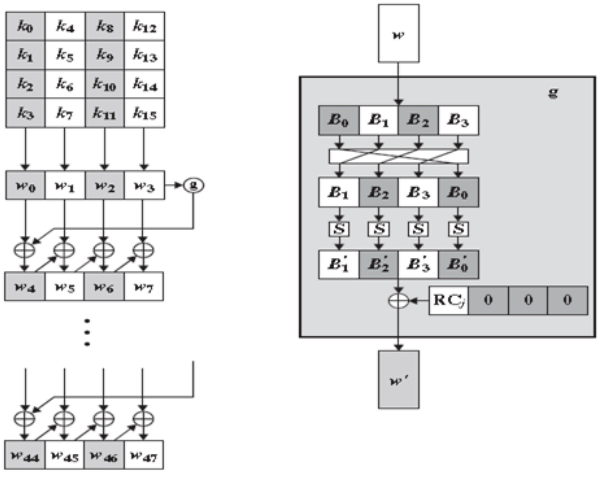
```
function HMAC(K, m)
  opad = [0x5c × blocksize]
  ipad = [0x36 × blocksize]
  if (length(K) > blocksize) then
    K = hash(K)
  end if
  for i from 0 to length(K) step 1
    ipad[i] ^= K[i]
    opad[i] ^= K[i]
  end for
  return hash(opad || hash(ipad || m))
```

## XIII. Thuật toán AES:

1) **Định nghĩa**: Thuật toán AES (Advanced Encryption Standard) là một thuật toán mã hóa khối (block cipher) đối xứng (symmetric) được sử dụng để mã hóa dữ liệu. Nó được chọn làm chuẩn mã hóa chính thức của chính phủ Hoa Kỳ vào năm 2001, thay thế cho chuẩn mã hóa trước đó là DES (Data Encryption Standard). AES là một thuật toán khối đối xứng, nghĩa là nó mã hóa các khối dữ liệu có cùng kích thước. Kích thước của khối là 128 bit và có thể được mã hóa bằng các khóa có độ dài khác nhau, từ 128 đến 256 bit.
2) **Cách hoạt động:** Thuật toán AES hoạt động bằng cách thực hiện một loạt các phép biến đổi trên khối dữ liệu đầu vào và khóa mã hóa. Đầu ra của thuật toán là khối dữ liệu được mã hóa.
3) **Ứng dụng:** AES được coi là một trong những thuật toán mã hóa đối xứng mạnh nhất hiện nay, được sử dụng rộng rãi trong các ứng dụng bảo mật dữ liệu như mã hóa tệp tin, mã hóa ổ đĩa, mã hóa mạng và mã hóa thông tin cá nhân trên các trang web.
4) **Mô tả thuật toán:**




Quy trình mã hóa



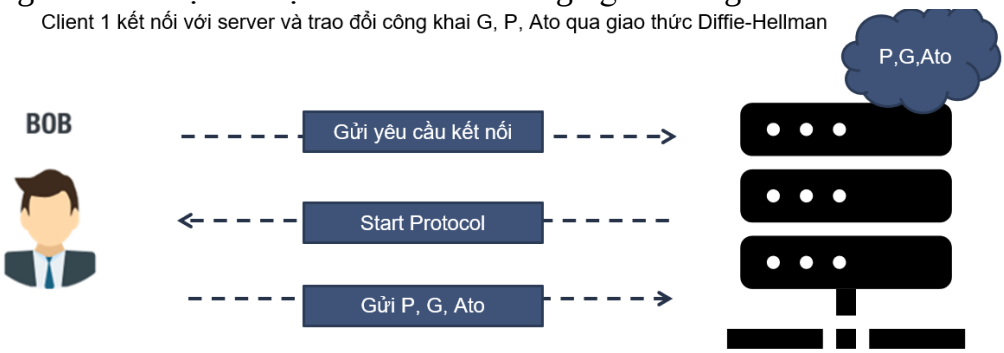
Quá trình mở rộng khóa

5) **Độ an toàn:** Độ an toàn của AES phụ thuộc vào kích thước khóa sử dụng. Kích thước khóa càng lớn thì khả năng tấn công và giải mã ngược trở nên khó khăn hơn. Hiện nay, AES với kích thước khóa 128 bit, 192 bit và 256 bit được sử dụng phổ biến nhất. Ngoài ra, độ an toàn của AES còn phụ thuộc vào cách thức triển khai thuật toán và cách sử dụng nó trong các ứng dụng bảo mật. Nếu triển khai không đúng cách hoặc sử dụng không đầy đủ các biện pháp bảo mật khác, thì việc sử dụng AES vẫn có thể bị đe dọa bởi các kỹ thuật tấn công khác.

## XIV. Mô tả ý tưởng:

❖ **Bước 1:** Tạo khóa công khai và khóa riêng tư cho mỗi người dùng

Sử dụng thuật toán Diffie-Hellman để tạo khóa công khai và khóa riêng tư cho mỗi người dùng. Mỗi người dùng sẽ giữ cho mình khóa riêng tư, trong khi khóa công khai của họ sẽ được chia sẻ với những người dùng khác.

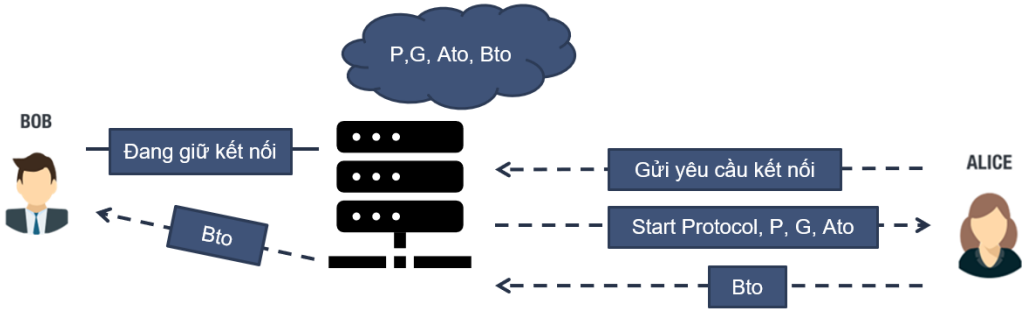



❖ **Bước 2:** Xác thực danh tính của các người dùng

Trước khi bắt đầu giao tiếp với nhau, các người dùng cần được xác thực danh tính của nhau. Sử dụng HMAC để xác thực danh tính của các người dùng. Mỗi người dùng sẽ gửi một chứng chỉ HMAC cho những người dùng khác để chứng minh rằng họ là người dùng hợp lệ.

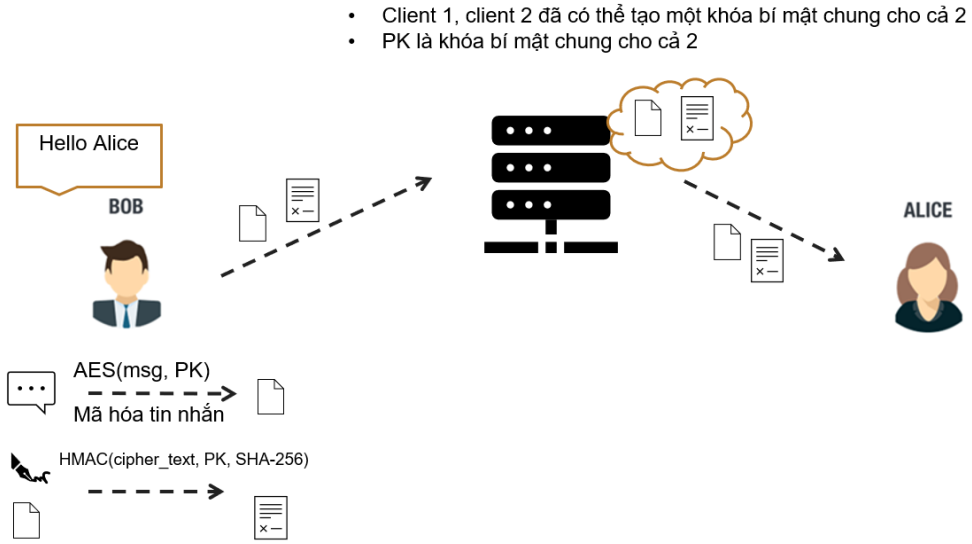
❖ **Bước 3:** Tạo khóa phiên

Sử dụng khóa công khai của các người dùng, sử dụng thuật toán Diffie-Hellman để tạo ra một khóa phiên bí mật. Khóa phiên này sẽ được sử dụng để mã hóa các tin nhắn giữa các người dùng.



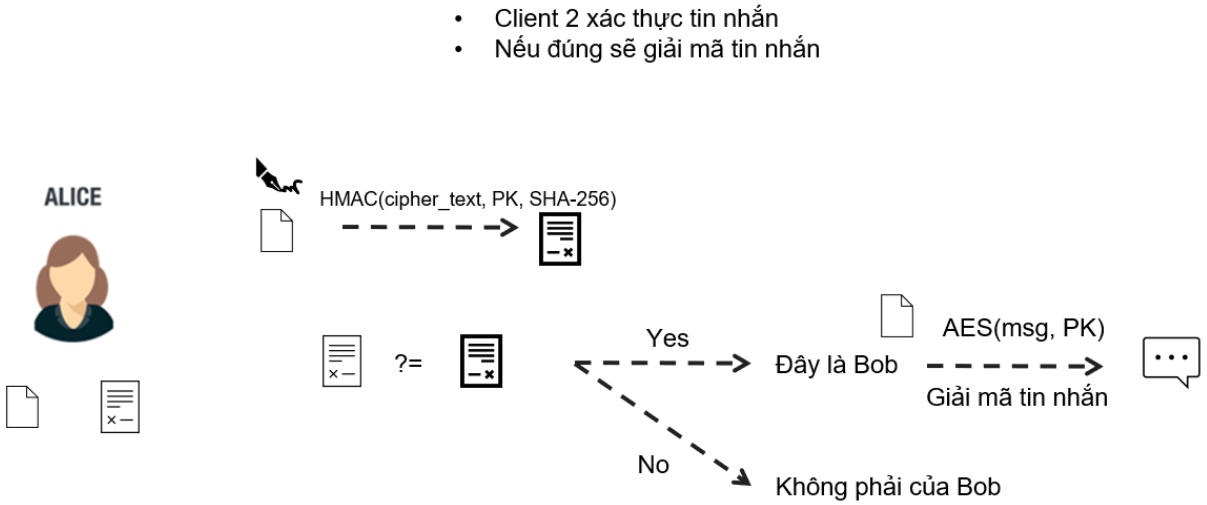

❖ **Bước 4:** Mã hóa và giải mã tin nhắn

Sử dụng thuật toán mã hóa đối xứng AES để mã hóa các tin nhắn giữa các người dùng bằng khóa phiên bí mật được tạo ra ở bước trước đó. Các tin nhắn này sau đó sẽ được gửi đi và nhận bởi các người dùng khác nhau.




❖ **Bước 5:** Xác thực tin nhắn

Sử dụng hash để xác thực tính toàn vẹn của tin nhắn khi nhận được. Nhận tin nhắn và sử dụng HMAC để tạo một mã băm, sau đó so sánh mã băm này với mã băm của tin nhắn gốc để đảm bảo rằng tin nhắn không bị thay đổi trong quá trình truyền tải.




➔ Với các bước trên, ta đã thiết kế một giao thức mã hóa đầu cuối sử dụng ZKP, Diffie-Hellman, AES và hash. Các thuật toán này sẽ giúp đảm bảo tính bảo mật và an toàn cho các thông tin được trao đổi giữa các người dùng.

## XV. Mô tả demo source code:

- B1: Chạy file Server.py, để khởi tạo Server Socket với HOST = '127.0.0.1' và PORT = 33000, lúc này Server sẽ lắng nghe kết nối từ Client
- B2: Cùng lúc chạy file Client.py để khởi tạo Client, một giao diện khung chat sẽ hiện ra và yêu cầu người dùng nhập tên để bắt đầu giao tiếp (VD: Alice)
- B3: Cùng lúc chạy file ClientCopy.py cũng như trên và nhập tên để giao tiếp (VD: Bob)
  B4: Sau khi cả hai Alice và Bob đã kết nối vào chung phòng chat thì Alice sẽ tạo ra 1 số bí mật **a** và Bob sẽ tạo ra 1 số bí mật **b**.
- B5: Tiếp đó, Client đầu tiên kết nối vào phòng chat ( ở đây là Alice ) sẽ tạo ra cặp số p ( là 1 số nguyên tố ), g ( là căn nguyên thuỷ ), sau đó tính $A = g^a \bmod p$. Rồi gửi p, g, A qua cho Bob.
- B6: Bob sẽ tính $B = g^b \bmod p$ và gửi B ngược lại cho Alice. Kế đó, Bob tính $s = A^b \bmod p$
- B7: Phía Alice sau khi nhận được B từ Bob gửi thì cũng tính $s = B^a \bmod p$
- B8: Cuối cùng, Alice và Bob đã tạo ra được 1 khoá bí mật chung s
- B9: Mỗi khi, Alice hoặc Bob gửi tin nhắn thì file Client.py/ClientCopy.py sẽ dùng thuật toán AES sử dụng key s để trao đổi qua cho nhau.

Trong tương lai sẽ có thể cải tiến và thêm các phương pháp khác.

Link video demo: https://www.youtube.com/watch?v=jyrVcCYvgus

## XVI. Nhận xét:

❖ **Ưu điểm:**

- Đem lại nhiều quyền riêng tư hơn. Mang lại cảm giác an toàn khi cần phải truyền đạt và lưu trữ thông tin nhạy cảm (chi tiết tài chính, y tế, tài liệu kinh doanh, thủ tục pháp lý…) trên không gian mạng. Một số ứng dụng thường được các doanh nghiệp lớn dùng như : Zalo, Lark, Skype, Telegram,...
- Đối với những tổ chức kinh tế, chính trị sẽ tránh được những thiệt hại khi những thông tin mật nếu vô tình bị lộ ra ngoài, cũng khó lòng giải mã ngay lập tức.
- Có thể chủ động sử dụng nó để giảm thiểu rủi ro khi giao tiếp trực tuyến trên các mạng xã hội như Facebook, Instagram, Twitter, … Ngày nay thì các ứng dụng mạng xã hội phổ biến đều có áp dụng chế độ Mã Hóa Đầu Cuối hoặc cần được người dùng bật lên khi muốn sử dụng

❖ **Nhược điểm:**

- Nếu mất khóa mã hóa, có thể sẽ mất quyền truy cập vào dữ liệu được nhận.
- Các mã khóa đơn giản không thể tránh khỏi việc bị tấn công và thử các khóa ngẫu nhiên cho đến khi khóa đúng được tìm thấy. Để giảm thiểu điều này có thể tăng chiều dài và độ phức tạp của khóa.

## XVII. Kết luận:

Mã Hóa Đầu Cuối là điểm cốt lõi cho trải nghiệm an toàn và riêng tư trên không gian mạng, mang lại nhiều lợi ích cho người dùng cũng như doanh nghiệp. Trong thời buổi hiện đại ngày nay, quyền riêng tư và ao toàn của người dùng trên không gian mạng đang rất nghiêm trọng. Mã Hóa Đầu Cuối là giải pháp có thể giải quyết vấn đề bảo mật, đảm bảo dữ liệu không bị tiết lộ khi máy chủ bị xâm nhập. Có thể nói, nếu thực hiện được đầy đủ các thuật toán đáng tin cậy, Mã Hóa Đầu Cuối sẽ có thể cung cấp mức độ an toàn cao nhất của việc bảo vệ dữ liệu.

## XVIII. Phân công công việc:

| MSSV | Họ và Tên | Công việc |
|---|---|---|
| 20127102 | Hoàng Hữu Minh An | Viết báo cáo, code demo, làm slide, quay video, thuyết trình. |
| 20127066 | Nguyễn Nhật Quân | Viết báo cáo, code demo, quay video |
| 20127192 | Nguyễn Anh Huy | Viết báo cáo, code demo, làm slide, quay video, thuyết trình. |
| 20127338 | Trương Gia Thịnh | Viết báo cáo, code demo, làm slide, quay video, thuyết trình |

## XIX. Các công việc đã hoàn thành

| STT | Các công việc đã hoàn thành | Mức độ hoàn thành |
|---|---|---|
| 1 | Viết báo cáo | 100% |
| 2 | Tìm hiểu Mã Hóa Đầu Cuối | 100% |
| 3 | Tìm hiểu AES | 100% |
| 4 | Tìm hiểu Diffie Hellman | 100% |
| 5 | Tìm hiểu TCP/IP Protocol | 100% |
| 6 | Tìm hiểu HMAC | 100% |
| 7 | Tìm hiểu ZKP | 50% |
| 8 | Khảo sát về Mã Hóa Đầu Cuối | 100% |
| 9 | Làm slide thuyết trình | 100% |
| 10 | Viết chương trình demo có ứng dụng nội dung nghiên cứu | 100% |
| 11 | Tìm hiểu cách sử dụng WireShark | 100% |
| 12 | Tìm tài liệu tham khảo | 100% |

## XX. Đánh giá và nhận xét quá trình làm việc:

Vì đây là một chủ đề ngoài, do đó trong quá trình tìm hiểu, nhóm cũng gặp một số khó khăn nhất định. Tuy nhiên nhóm vẫn tuân thủ và hoàn thành theo phương hướng và kế hoạch đã đề ra ban đầu. Trong lúc tìm hiểu, nhóm cũng cảm thấy một số kiến thức và mô hình mang tính khả thi không cao. Có thể kể đến như mô hình ZKP, một mô hình đòi hỏi khá nhiều về kiến thức mã hóa và bảo mật nâng cao tương đối khó triển khai trong sản phẩm demo nhỏ thực tế lúc này. Do do nhóm đã phải điều chỉnh một số mô hình để giúp cho phù hợp hơn với nội dung. Bên cạnh

đó, việc sắp xếp thời gian làm việc nhóm cũng gặp đôi chút khó khăn. Tuy nhiên, sau cùng nhóm cũng đã hoàn thành được nội dung tìm hiểu và có được sản phẩm demo thực tế tương ứng.

Nhìn chung, nhóm cảm thấy đây là một chủ đề rất hay và mang tính thực tế cao. Trong tương lai, có thể sẽ ứng dụng rộng rãi hơn và hoàn toàn có thể cải tiến và tích hợp thêm được các công nghệ mới.

## XXI. Tài liệu tham khảo:

[1]: End-to-End Encryption Explained – This is How Every Messenger Should Encrypt Data - YouTube

[2]:End to End Encryption (E2EE) - Computerphile - YouTube

[2]:How To Create A Real Time Chat App In Python Using Socket Programming | Part 1 - YouTube

[3]:Python Live Chat Room Tutorial Using Flask & SocketIO - YouTube

[4]:End-to-End Encrypted Chat with JS & Web Crypto API (getstream.io)

[5]:end-to-end-encryption · GitHub Topics

[6]:end_to_end_encryption/api.md at master · nextcloud/end_to_end_encryption (github.com)

[7]:Use end-to-end encryption for Teams meetings - Microsoft Support

[8]Automated Symbolic Verification of Telegram's MTProto 2.0 | Papers With Code

[9]Blockchain-Enabled End-to-End Encryption for Instant Messaging Applications | Papers With Code

[10] On End-to-End Encryption: Britta Hale and Chelsea Komlo

[11]Improving Non-Experts' Understanding of End-to-End Encryption: An Exploratory Study

[12] End-to-End Encryption in Messaging Services and National Security—Case of WhatsApp Messenger

Cảm ơn quý thầy cô và các bạn đã xem.