Create by Lê Trọng Nghĩa

PRIME AND DIFFIE-HELLMAN KEY EXCHANGE

IC G14 PJRP-001

DIFFIE-HELLMAN KEY EXCHANGE, LARGE NUMBER PROCESSING SYSTEM

The report is part of the first group project of the Introduction to Cryptography course, a module provided by the University of Natural Sciences, VNU-HCM.

CQ2022/22, Introduction to Cryptography, University of Natural Sciences, VNU-HCM

INTRODUCTION TO CRYTOGRAPHY

A PROJECT REPORT

IC G14 PJRP-001

RESPONSIBILITY

GROUP 14

Member 01: 22120257

Đinh Lê Gia Như

Member 02: 22120226

Lê Trọng Nghĩa

Member 03: 22120192

Nguyễn Đăng Long

FACULTY OF INFORMATION TECHNOLOGY

UNIVERSITY OF SCIENCE, VNU-HCM CITY UNIVERSITY

HO CHI MINH CITY, VIET NAM COUNTRY

October, 2024

Language: English, Vietnamese

Tóm tắt

Tài liệu IC G14 PJRP-001 này là tài liệu báo cáo thuộc Đồ án Lab01 (Dự án nhỏ Lab01), khóa học Nhập môn mã hóa – mật mã, bộ môn Công nghệ trí thức, Trường Đại học Khoa học Tự Nhiên, Đại học Quốc Gia Thành phố Hồ Chí Minh, Việt Nam.

IC G14 PJRP-001 báo cáo các vấn đề về giao thức trao đổi khóa Diffie-Hellman, và hệ thống xử lý số nguyên lớn được áp dụng vào dự án Lab01 này.

Các tài liệu, văn bản có liên quan đến Đồ án Lab01 bao gồm: Báo cáo IC G14 PJRP-001, Báo cáo IC G14 PJRP-002, Báo cáo IC G14 PJRP-003 được lưu trữ trong thư mục project_01_report và nộp bài tập đồ án trên trang môn học (Moodle) của Khoa Công nghệ thông tin, Trường Đại học Khoa học Tự nhiên, Đại học Quốc gia Thành phố Hồ Chí Minh, Việt Nam. Ngoài ra, mã nguồn của đồ án đã thực hiện được lưu trữ trong thư mục project_01_source có kèm theo với các tài liệu báo cáo này khi kết thúc đồ án.

Từ khóa có liên quan: diffie-hellman; key exchange; NIST DH; cryptography; VPN; AES; TLS/SSL; IPsec; ECDH; TLS/HTTPS

Lời cảm ơn

Người thực hiện bài báo cáo này, tôi là Lê Trọng Nghĩa, trân trọng ghi nhận đóng góp và đánh giá cao đến những đóng góp của các cá nhân Đinh Lê Gia Như, Lê Trọng Nghĩa, Nguyễn Đăng Long là thành viên của nhóm 14 thực hiện đồ án Lab01 có các nội dung liên quan đến các vấn đề được nêu có trong tài liệu này bao gồm các vấn đề về công nghệ thông tin, mật mã và bảo mật.

Tác giả bài báo cáo cũng xin cảm ơn các hướng dẫn, gợi ý về đồ án, và các giải đáp thắc mắc từ phía giảng viên phụ trách khóa học, người quản lý dự án có liên quan.

Nguồn tham khảo

Nguồn tài liệu tham khảo:

a. Tiêu chuẩn hóa: RFC 2631, NIST DH, NIST SP 800-56, NIST SP 800-131A Re.2
RFC 2631: https://datatracker.ietf.org/doc/html/rfc2631
NIST DH: https://csrc.nist.gov/glossary/term/dh
NIST SP 800-56: https://csrc.nist.gov/files/pubs/sp/800/56/a/upd1/final/docs/sp800-56-draft-jul2005.pdf
NIST SP 800-131A Re.2: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar2.pdf

b. Diễn đàn: Wolfram MathWorld, Wikipedia, GeeksforGeeks
Wolfram MathWorld: https://mathworld.wolfram.com/Diffie-HellmanProtocol.html
Wikipedia: https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange
GeeksforGeeks: https://www.geeksforgeeks.org/implementation-diffie-hellman-algorithm/

c. Tài liệu tham khảo:
An Introduction to Mathematical Cryptography Book.

Table of Contents

1. Giao thức Diffie – Hellman

1.1 Tổng quan

Giao thức Diffie-Hellman là một phương pháp để hai người dùng máy tính tạo ra một khóa riêng được chia sẻ mà sau đó họ có thể trao đổi thông tin qua một kênh không an toàn. Giả sử những người dùng này có tên là Alice và Bob. Đầu tiên, họ thống nhất về hai số nguyên tố g và p, trong đó p lớn (thường là ít nhất 512 bit) và g là một căn nguyên thủy modulo p. (Trong thực tế, tốt nhất là chọn p sao cho $\frac{p-1}{2}$ cũng là số nguyên tố.) Các số g và p không cần phải được giữ bí mật với những người dùng khác. Bây giờ Alice chọn một số ngẫu nhiên lớn a làm khóa riêng của mình và Bob cũng chọn một số lớn b. Sau đó, Alice tính $A = g^a \ (mod\ p)$ mà cô ấy gửi cho Bob, và Bob tính $B = \ g^b \ (mod\ p)$, mà anh ấy gửi cho Alice.

Bây giờ cả Alice và Bob đều tính toán khóa chung của họ $K = \ g^{ab} \ (mod\ p)$, mà Alice tính toán như sau

$$K = \ B^a \ (mod\ p) = (g^b)^a \ (mod\ p)$$

và Bob tính toán như sau

$$K = \ A^b \ (mod\ p) = (g^a)^b \ (mod\ p)$$

Alice và Bob hiện có thể sử dụng khóa chung K của họ để trao đổi thông tin mà không phải lo lắng về việc những người dùng khác có được thông tin này. Để một kẻ nghe lén tiềm năng (Eve) có thể làm như vậy, trước tiên cô ta cần phải có được $K = \ g^{ab} \ (mod\ p)$ chỉ biết g, p, $A = \ g^a \ (mod\ p)$ và $B = \ g^b \ (mod\ p)$.

Điều này có thể được thực hiện bằng cách tính a từ $A = \ g^a \ (mod\ p)$ hoặc b từ $B = \ g^b \ (mod\ p)$. Đây là bài toán logarit rời rạc, không khả thi về mặt tính toán đối với p lớn. Việc tính logarit rời rạc của một số modulo p mất khoảng thời gian gần bằng thời gian phân tích tích của hai số nguyên tố có cùng kích thước với p, đó là điều mà hệ thống mật mã RSA dựa vào. Do đó, giao thức Diffie-Hellman gần như an toàn như RSA.

1.2 Ứng dụng

Giao thức Diffie-Hellman (DH) là một trong những phương pháp đầu tiên giúp hai bên có thể trao đổi khóa bí mật qua một kênh không an toàn. Điều này đặc biệt quan trọng trong môi trường mạng, nơi mà các kênh truyền tải dữ liệu dễ bị nghe lén hoặc tấn công. Giao thức này tận dụng các phép toán số học để tạo ra một khóa chung duy nhất cho hai bên, giúp bảo mật thông tin mà không cần trao đổi khóa trực tiếp.

1.2.1 Trao Đổi Khóa An Toàn Trong Môi Trường Mạng

Diffie-Hellman là một trong những giao thức chính cho việc trao đổi khóa trong nhiều hệ thống bảo mật. Khóa chung tạo ra nhờ giao thức DH có thể được sử dụng cho các thuật toán mã hóa đối xứng (chẳng hạn như AES). Điều này đảm bảo rằng dữ liệu trao đổi được mã hóa và chỉ có thể giải mã bởi hai bên tham gia.

1.2.2 Xây Dựng Cơ Sở Cho Các Giao Thức Bảo Mật Khác

Diffie-Hellman là nền tảng cho nhiều giao thức và hệ thống bảo mật quan trọng, bao gồm:

- **TLS/SSL**: Được sử dụng để bảo mật kết nối giữa máy khách và máy chủ trong các giao dịch qua internet. TLS/SSL sử dụng DH để trao đổi khóa phiên, tạo nền tảng bảo mật cho các trang web qua giao thức HTTPS.
- **IPsec**: Giao thức bảo mật mạng IPsec sử dụng DH trong quá trình thiết lập kênh bảo mật giữa các thiết bị, tạo ra môi trường an toàn cho truyền tải dữ liệu trong VPN (Virtual Private Network).
- **Tin nhắn mã hóa đầu cuối**: Nhiều ứng dụng nhắn tin như Signal và WhatsApp sử dụng DH để thiết lập khóa phiên cho mỗi lần trò chuyện. Điều này đảm bảo rằng nội dung tin nhắn chỉ có thể được đọc bởi người gửi và người nhận.

1.2.3 Bảo Mật Thông Tin Cá Nhân Trong Các Ứng Dụng Số

Trong các hệ thống yêu cầu bảo mật thông tin cá nhân như dịch vụ ngân hàng số, giao thức DH cho phép thiết lập các khóa bí mật mà không yêu cầu phải lưu trữ khóa trên máy chủ. Điều này giảm nguy cơ thông tin cá nhân bị lộ khi các hệ thống bị xâm phạm.

1.2.4 Phát Triển Thành Các Biến Thể Bảo Mật Cao Hơn

Giao thức Diffie-Hellman đã được phát triển thành các biến thể như **Elliptic Curve Diffie-Hellman (ECDH)** để tăng cường tính an toàn và hiệu quả. Các biến thể này ứng dụng lý thuyết đường cong elliptic, cho phép đạt được cùng mức độ bảo mật với độ dài khóa ngắn hơn, phù hợp cho các thiết bị có giới hạn tài nguyên như điện thoại di động.

☞ Giao thức Diffie-Hellman đóng vai trò quan trọng trong bảo mật truyền thông và trao đổi dữ liệu an toàn. Với khả năng tạo khóa bí mật mà không cần kênh bảo mật, DH đã trở thành nền tảng cho nhiều giao thức và ứng dụng bảo mật phổ biến hiện nay, bảo vệ thông tin cá nhân và duy trì tính toàn vẹn của dữ liệu.

1.3 TLS/HTTPS

TLS (Transport Layer Security) là giao thức bảo mật được sử dụng rộng rãi để mã hóa dữ liệu truyền giữa các trình duyệt và máy chủ web, giúp bảo vệ dữ liệu nhạy cảm trước các cuộc tấn công đánh cắp dữ liệu và nghe lén. HTTPS (HTTP Secure) là phiên bản an toàn của HTTP, trong đó TLS đóng vai trò bảo vệ kênh truyền dữ liệu.

1.4 TLS/HTTPS với giao thức Diffie – Hellman

Diffie-Hellman được sử dụng trong TLS/HTTPS để thực hiện trao đổi khóa bảo mật giữa trình duyệt và máy chủ web, đảm bảo rằng chỉ có hai bên tham gia vào phiên truyền dữ liệu mới có thể giải mã thông tin của nhau.

1.4.1 Cách hoạt động

1. **Bắt đầu phiên TLS**: Khi một người dùng mở trang web HTTPS, trình duyệt của người dùng sẽ gửi yêu cầu đến máy chủ để bắt đầu kết nối TLS.

2. **Xác thực và lựa chọn thuật toán mã hóa**: Máy chủ phản hồi với chứng chỉ số và danh sách các thuật toán mã hóa mà nó hỗ trợ. Trình duyệt sẽ kiểm tra tính hợp lệ của chứng chỉ và chọn một bộ mã hóa hỗ trợ Diffie-Hellman để trao đổi khóa an toàn.

3. **Trao đổi khóa Diffie-Hellman**:
    - Cả trình duyệt và máy chủ cùng thỏa thuận một số nguyên tố lớn p và một cơ sở g.
    - Trình duyệt chọn khóa riêng ngẫu nhiên a và tính khóa công khai $A = g^a \mod p$.
    - Máy chủ chọn khóa riêng b và tính khóa công khai $B = g^b \mod p$.
    - Trình duyệt gửi A cho máy chủ và máy chủ gửi cho trình duyệt.
4. **Tính toán khóa chung**: Cả hai bên sử dụng khóa công khai của nhau để tính toán khóa chung:
    - Trình duyệt tính $s = B^a \mod p$.
    - Máy chủ tính $s = A^b \mod p$.
    - Kết quả s là một khóa phiên bí mật dùng để mã hóa tất cả dữ liệu được trao đổi trong phiên TLS.
5. **Mã hóa dữ liệu**: Sau khi khóa phiên được thiết lập, trình duyệt và máy chủ bắt đầu mã hóa và giải mã các thông điệp trao đổi trong suốt phiên giao dịch.

1.4.2 Lợi ích

- **Bảo mật chống lại tấn công nghe lén**: Diffie-Hellman tạo khóa phiên mới cho mỗi kết nối, ngăn chặn kẻ tấn công đoán được khóa mã hóa của phiên khác.
- **Không yêu cầu lưu trữ khóa bí mật lâu dài**: Khóa phiên chỉ tồn tại trong thời gian phiên làm việc, vì vậy khi phiên kết thúc, dữ liệu mã hóa không thể giải mã ngay cả khi kẻ tấn công lấy được khóa công khai.
- **Khả năng bảo mật về sau**: Khóa phiên của từng phiên là duy nhất. Điều này giúp bảo vệ dữ liệu quá khứ khỏi các tấn công giải mã trong tương lai.
-

2 Hệ thống xử lý số nguyên lớn

2.1 Giới thiệu (class BigInt512)

BigInt512 là 1 class biểu diễn số nguyên lớn, tối thiểu 512 bit. Class này lưu trữ giá trị theo từng phần 64 bit và hỗ trợ các toán tử phù hợp với yêu cầu cài đặt.

2.2 Mô tả

Cấu trúc của BigInt512: Sử dụng 1 mảng 8 phần tử (parts[8]) thuộc kiểu uint64_t để lưu trữ số lớn dưới dạng các khối 64 bit. Mỗi phần tử trong mảng biểu diễn 64 bit của số lớn, và tổng cộng 8 phần tử có khả năng lưu trữ 512 bit (8 × 64 bit). Xem mô tả cụ thể hơn ở mục 1. IC G14 PJRP-003.

Phương thức khởi tạo BigInt512:

- Constructor mặc định: Khởi tạo một đối tượng BigInt512 với giá trị ban đầu là 0 bằng cách đặt tất cả các phần tử của parts về 0.
- Constructor từ chuỗi hex: Cho phép khởi tạo BigInt512 từ một chuỗi biểu diễn hex. Chuỗi này được điều chỉnh để có độ dài 128 ký tự (512 bit), sau đó chia thành các nhóm 16 ký tự (64 bit) để lưu trữ vào mảng parts.

2.3 Điểm tối ưu

Tối ưu hóa bằng cách chia phần 64 bit: BigInt512 chia số lớn thành 8 phần 64 bit, giúp cho các phép toán trên từng phần dễ dàng và hiệu quả hơn. Điều này cho phép tận dụng các toán tử cơ bản của uint64_t (được xử lý nhanh bởi CPU) mà không cần phải thao tác với từng bit riêng lẻ.

Dễ mở rộng và bảo trì: BigInt512 sử dụng cấu trúc mảng cố định và các phép toán cơ bản, giúp mở rộng thêm các toán tử và tính năng (như trừ, nhân, chia) một cách dễ dàng mà không làm phức tạp cấu trúc.

2.4 Nguyên lý làm việc

Lưu trữ số lớn: BigInt512 lưu trữ giá trị số dưới dạng 8 phần tử uint64_t trong mảng parts[8], mỗi phần tử chứa 64 bit, tổng cộng 512 bit.

Khởi tạo từ chuỗi hex: Khi khởi tạo từ một chuỗi hex, mỗi 16 ký tự trong chuỗi (đại diện cho 64 bit) được chuyển thành một số uint64_t và lưu vào phần tử tương ứng của mảng parts. Như vậy, chuỗi hex đầu vào có độ dài tối đa 128 ký tự (512 bit) sẽ được phân bố đều trong mảng parts.