GIẢI PHÁP XÁC THỰC VÀ BẢO MẬT DỮ LIỆU TRUYỀN, CHỐNG TẤN CÔNG LƯỢNG TỬ VÀ MAN-IN-THE-MIDDLE

Đinh Xuân Lâm

Khoa Công nghệ thông tin, Trường Đại học Ngoại ngữ -Tin học TP.HCM

lamdx@huflit.edu.vn

`

TÓM TẮT— Cơ chế trao đổi khóa là điều kiện tiên quyết để đảm bảo tính xác thực và bảo mật kết nối client-server. Kỹ thuật mã hóa dữ liệu truyền thống dựa trên kiến trúc PKI không còn là giải pháp đảm bảo an toàn tuyệt đối cho hoạt động trao đổi khóa và hoạt động truyền dữ liệu qua mạng, đặc biệt với sự xuất hiện của các máy tính lượng tử. Nghiên cứu phân tích lỗ hổng và rủi ro tấn công lượng tử của giao thức trao đổi khóa dựa trên các hệ mật khóa công khai như Diffie Hellman, từ đó đề xuất một giải pháp trao đổi khóa theo mô hình đường ống mã hóa AES, Diffie Hellman cải biên có xác thực, kết hợp giấu tin, có khả năng chống tấn công trung gian và hạn chế rủi ro bảo mật từ năng lực lượng tử, giúp giảm nguy cơ rò rỉ, giả mạo thông tin, gia tăng tính an toàn cho quá trình xác thực và truyền dữ liệu trong các ứng dụng client-server. Nghiên cứu cũng cung cấp một cái nhìn tổng quan về hiệu năng và khả năng triển khai thực tế giải pháp từ các kết quả thực nghiệm.

Từ khóa— Bảo mật client-server, ẩn giấu thông tin, giao thức trao đổi khóa, ngăn chặn tấn công man-in-the-middle, truyền thông chống lượng tử

I. GIỚI THIỆU

Hiện nay, kiến trúc client-server được ứng dụng rộng rãi trong môi trường web, dịch vụ đám mây, ứng dụng di động và IoT. L. Zhang đã chỉ ra rằng mọi ứng dụng thanh toán hiện đại trong thương mại điện tử đều dựa trên mô hình client-server, trong đó xác thực và bảo mật dữ liệu truyền đóng vai trò then chốt để đảm bảo giao dịch an toàn và hợp pháp [1]. Nhiều nghiên cứu chỉ ra rằng cơ chế trao đổi khóa là điều kiện tiên quyết để đảm bảo tính bảo mật, xác thực với khả năng chống tấn công trung gian. Nghiên cứu của nhóm tác giả Saru Kumari nhấn mạnh vai trò bắt buộc của trao đổi khóa để chống lại các lỗ hổng trong xác thực [2].

Các giao thức trao đổi khóa được sử dụng trong chức năng xác thực hệ điều hành, bảo mật các dịch vụ windows và các ứng dụng client-server hiện nay phổ biến là ECDHE trong TLS 1.3, IKEv2 trong IPsec VPN, SSH key exchange. Điểm chung của các giao thức này là đều sử dụng kiến trúc PKI trong quá trình trao đổi khóa.

Sự xuất hiện của các máy tính lượng tử làm cho các kỹ thuật mã hóa khóa công khai truyền thống không còn đảm bảo an toàn tuyệt đối, chẳng hạn như với RSA/ECC, hai trụ cột của xác thực và trao đổi khóa trong PKI. Nghiên cứu của nhóm tác giả Gupta, R., & Sharma, N. phân tích nguy cơ từ máy tính lượng tử đối với các hệ mật khóa công khai, đe dọa TLS và PKI trong mô hình client-server [3]. Các tác giả J. Chen, L. Wang đã phân tích logarit rời rạc và nguy cơ lượng tử đối với thuật toán trao đổi khóa Diffie Hellman/ECDH [4]. Thuật toán của Shor cho phép phân tích nhanh các số nguyên lớn thành thừa số nguyên tố và tính toán nhanh logarit rời rạc, từ đó tính được khóa bí mật từ khóa công khai [3]. Khi năng lực lượng tử đủ mạnh, các máy tính lượng tử có thể bẻ khóa các hệ mật khóa công khai như RSA/ECC/Diffie Hellman trong thời gian thực.

Thời điểm mà năng lực lượng tử đủ mạnh để có thể phá vỡ các thuật toán mã hóa khóa công khai hiện nay được dự báo là trong thập niên 2030. Báo cáo NIST 8547 đề xuất thời hạn ngừng sử dụng các thuật toán RSA-2048 và ECC-256 vào năm 2030 và cấm hoàn toàn RSA và ECC vào năm 2035 [5]. Các lược đồ thiết lập khóa có lỗ hổng lượng tử và thời gian ngừng sử dụng hay loại bỏ được NIST khuyến nghị được trình bày trong bảng 1. Do đó, xu hướng thế giới là chuyển đổi sang các giải pháp thiết lập và phân phối khóa mới để bảo vệ dữ liệu trong tương lai.

Bảng 1. *Danh sách các lược đồ thiết lập khóa có lỗ hổng lượng tử*

Lược đồ thiết lập khóa	Các tham số	Thời gian chuyển đổi
Finite field DH và MQV	112 bits of security strength	Ngừng sử dụng 2030, loại bỏ 2035
	≥ 128 bits of security strength	Loại bỏ 2035
ECDH và MQC	112 bits of security strength	Ngừng sử dụng 2030, loại bỏ 2035
	≥ 128 bits of security strength	Loại bỏ 2035
RSA	112 bits of security strength	Ngừng sử dụng 2030, loại bỏ 2035
	≥ 128 bits of security strength	Loại bỏ 2035
Nguồn: https://doi.org/10.6028/NIST.IR.8547.ipd		

Một số giải pháp xác thực, mã hóa hậu lượng tử và phân phối khóa lượng tử đã được đề xuất và đang được thử nghiệm trong môi trường client - server.

Nghiên cứu này đề xuất một giải pháp trao đổi khóa và một mô hình đường ống kết hợp kỹ thuật mã hóa AES-256, Diffie Hellman cải biên với kỹ thuật giấu tin, trong đó mã hóa để bảo mật nội dung và kỹ thuật giấu tin để che giấu sự tồn tại của dữ liệu truyền, nhằm giảm rủi ro lộ thông tin khi bị tấn công lượng tử hoặc tấn công man-in-the-miidle. Bên cạnh đó, nghiên cứu cũng tập trung phân tích kết quả thực nghiệm về hiệu suất khi triển khai giải pháp đề xuất trong môi trường mô phỏng.

Bài báo gồm 5 phần: phần I giới thiệu tổng quan nguyên nhân phát sinh ý tưởng nghiên cứu. Phần II giới thiệu các nghiên cứu liên quan và khả năng triển khai thực tế. Phần III giới thiệu phương pháp nghiên cứu, ý tưởng thiết kế và giải pháp đề xuất. Phần 4 trình bày kết quả thực nghiệm và phân tích về độ bảo mật và hiệu năng của giải pháp đề xuất. Cuối cùng phần 5 là kết luận và hướng triển khai thực tế.

II. CÁC CÔNG TRÌNH LIÊN QUAN

Owolabi và Elly đề xuất các chiến lược tối ưu hóa cho giao thức trao đổi khóa lai, một phương pháp kết hợp giữa mật mã truyền thống ECDHE và mật mã hậu lượng tử để chống tấn công lượng tử [6]. Nhưng khi triển khai thực tế vẫn tồn tại hạn chế chi phí tính toán cao, kích thước khóa lớn, khó tích hợp với các chuẩn hiện có.

Nguyen, T. H., & Pham, Q. A. đề xuất giải pháp loại bỏ mật khẩu bằng cách kết hợp mã hóa bất đối xứng với dấu vân tay thiết bị, cơ chế giúp định danh duy nhất người dùng mà không cần lưu trữ bất kỳ chuỗi ký tự nào trên máy chủ [7]. Tuy nhiên, khi triển khai thực tế, giải pháp gặp nhiều hạn chế về chi phí tính toán và quản lý khóa, độ tin cậy của công nghệ nhận diện thiết bị chưa tuyệt đối, gây khó khăn cho khả năng mở rộng hệ thống.

Suman Bhoi đề xuất giải pháp sử dụng mật khẩu đa ngôn ngữ kết hợp các bảng chữ cái khác nhau như latin, kanji, cyrillic, làm tăng độ khó cho các công cụ bẻ khóa lên gấp hàng chục nghìn lần so với mật khẩu chỉ dùng tiếng Anh [8]. Tuy nhiên, khi triển khai thực tế, nghiên cứu tồn tại nhiều hạn chế về vấn đề tương thích hệ thống và rào cản người dùng.

Song, J., Han, J., Au, M. H., Yang, R., & Sun, C. nghiên cứu về các thuật toán xác thực dựa trên lưới, để bảo vệ mật khẩu và chữ ký số trước sức mạnh của máy tính lượng tử trong tương lai [9]. Nghiên cứu còn những hạn chế về hiệu năng, kích thước khóa lớn, khó quản lý động, khó tích hợp vào hệ thống thực tế.

Năm 2024, NIST chuẩn hóa Kyber thành nền tảng trao đổi khóa hậu lượng tử. Kyber cùng với Dilithium, McEliece trở thành các nền tảng thay thế RSA/ECC trong các hệ thống bảo mật toàn cầu ở kỷ nguyên hậu lượng tử [3, 10]. Tuy nhiên, chi phí chuyển đổi sang hệ thống hậu lượng tử rất lớn, do phải thay đổi toàn bộ hạ tầng PKI. Đặc biệt trong môi trường IoT công nghiệp, có số lượng thiết bị lớn, nhưng đa số là các hệ thống nhúng có tài nguyên tính toán hạn chế, nên cần tối ưu hóa việc tiêu thụ năng lượng và tài nguyên tính toán khi triển khai.

Đa số nghiên cứu tập trung vào việc sử dụng chữ ký số hay sinh trắc học kết hợp với hệ mật khóa công khai để bảo mật tiến trình xác thực trước sức mạnh của máy tính lượng tử, Nhưng khi triển khai thực tế thì đa số gặp trở ngại về vấn đề đồng bộ thiết bị người dùng, độ tin cậy của chứng chỉ số và vấn đề chi phí. Ngoài ra, các nghiên cứu cũng ít phân tích thực nghiệm về hiệu năng và khả năng vượt xác thực đa lớp không cần mật khẩu của người dùng [11].

III. PHƯƠNG PHÁP LUẬN VÀ GIẢI PHÁP ĐỀ XUẤT

Nghiên cứu này đề xuất một cơ chế trao đổi khóa để bảo vệ mật khẩu xác thực, sử dụng AES và kỹ thuật giấu tin để hạn chế rủi ro tấn công lượng tử, Diffie Hellman cải biên có chức năng xác thực để chống tấn công trung gian, nhưng vẫn đảm bảo hiệu năng hệ thống. Giải pháp có khả năng bảo mật chuyển tiếp, hạn chế rủi ro "thu thập dữ liệu trước, giải mã dữ liệu sau khi máy tính lượng tử đủ mạnh".

Giải pháp đề xuất được xây dựng dựa trên việc nghiên cứu tổng quan tài liệu, triển khai thực nghiệm trong môi trường mô phỏng, phân tích và đánh giá kết quả thực nghiệm so với các giải pháp trao đổi khóa dựa trên RSA/ECC/Diffie Hellman.

Các máy tính lượng tử không bẻ khóa bằng sức mạnh tính toán thuần túy mà bằng khả năng giải quyết các bài toán phân tích thừa số nguyên tố và logarit rời rạc trong thời gian đa thức. Để bẻ khóa RSA-2048, máy tính lượng tử cần khoảng 20 triệu Qubit, nhưng chỉ cần 13 triệu Qubit để bẻ khóa ECC-256. Khi đạt tới ngưỡng cần thiết, RSA/ECC sẽ bị bẻ khóa gần như ngay lập tức.

Với hệ mật đối xứng AES, thuật toán Grover cho phép máy tính lượng tử bẻ khóa AES với độ phức tạp giảm xuống còn $\sqrt{N}$. Nên AES-256 vẫn được xem là an toàn trước các tấn công lượng tử. NIST xếp AES-256 vào nhóm thuật toán có khả năng bảo mật chống lượng tử cao nhất [5].

Thông tin serial number của máy client và mật khẩu người dùng (password) lưu tại ứng dụng server dưới dạng mã băm theo từng username khi đăng ký người dùng mới.

H1 = SHA1(password), H2 = SHA256(password), H3 = SHA256(serial number)

A. TIẾN TRÌNH XÁC THỰC GIỮA ỨNG DỤNG CLIENT VÀ ỨNG DỤNG SERVER

1. Kết nối ứng dụng client với ứng dụng server
2. Người dùng nhập username và mật khẩu đăng nhập pass tại ứng dụng client bằng máy đã đăng ký.
3. Ứng dụng client tính:
   H4 = SHA1(pass), key = SHA256(pass)
   data = H4 & H3
   Phát sinh các số 16 bytes ngẫu nhiên salt và iv
   Tạo khóa K = PBKDF2(key, salt, 32, interations)
   Mã hóa data với khóa K: cipher = $AES256_K$(data)
   Gửi username, cipher, salt, iv, interations cho ứng dụng server
   Chờ nhận thông tin từ ứng dụng server
   Nếu nhận "close", client đóng kết nối. Kết thúc tiến trình xác thực
   Ngược lại, nhận giá trị x
      Phát sinh a, c, n từ key
      Tính y = a^c mod n
      Gửi y cho server
      Tính z = x^c mod n
      Tính passkey = SHA256(z)
      Dùng passkey để mã hóa/giải mã AES256 dữ liệu truyền trong phiên đăng nhập
4. Ứng dụng server
   Nhận và kiểm tra sự tồn tại của username
   IF exist(username)
   datasrv = H1 & H3, keysrv = H2
   Nhận cipher, salt, iv, interactions từ client
   Tạo khóa K = PBKDF2(keysrv, salt, 32, interations)
   Giải mã cipher với khóa K: data2= $AES256_K$(cipher)
   IF LEFT (data2, 20 bytes) == H1:
      IF RIGHT (data2, 32 bytes) == H3:
         Phát sinh a, b, n từ H2
         Tính x = a^b mod n, gửi x cho client
         Nhận y từ client tính z = y^b mod n
         Tính passkey = SHA256(z)
         Client dùng passkey để mã hóa/giải mã AES256 dữ liệu truyền trong phiên đăng nhập
   ELSE:
      Send yêu cầu tiến trình xác thực bổ sung cho client như nhận dạng vân tay, khuôn mặt, do máy xác thực không phải là máy đăng ký ban đầu.
   ELSE:
      Send "close" cho client và đóng phiên kết nối

B. TIẾN TRÌNH TRAO ĐỔI DỮ LIỆU GIỮA CLIENT VÀ SERVER TRONG PHIÊN KẾT NỐI

Bên gửi:

1. Phát sinh các số 16 bytes ngẫu nhiên **salt** và **iv**
2. K = PBKDF2(passkey, salt, 32, **interations**): dùng PBKDF2 tạo khóa mã hóa từ passkey và salt, với số lần băm interations.
3. **payload** = $AES256_K$(m): mã hóa AES thông điệp gửi **m** với khóa K, mode cbc.
4. size = kích thước payload
5. hinhemb = embbed (hinhbmp, payload, size, seed) : nhúng nội dung **payload** vào các bytes trong hình BMP theo phương pháp trọng số thấp LSB, có xáo trộn thứ tự nhúng các byte trong hình tùy thuộc giá trị **seed** 4 bytes ngẫu nhiên.
6. **seedenc** = $AES256_K$(seed), **sizeenc** = $AES256_K$(size): mã hóa AES giá trị seed và size với khóa K, mode cbc.

7. Gửi hinhemb kèm salt, iv, interations, sizeenc và seedenc cho bên nhận

Bên nhận:

1. Nhận tệp hình hinhemb và các giá trị salt, iv, interations, seedenc, sizeenc
2. K = PBKDF2(passkey, salt, 32, **interations**): tạo khóa giải mã từ passkey và salt.
3. Giải mã seed = $AES256_K$(seedenc), size = $AES256_K$(sizeenc)
4. payload = extract (hinhemb, size, seed): trích xuất dữ liệu từ tệp hình hinhemb
5. m = $AES256_K$(payload): giải mã payload để thu được thông điệp m

C. THUẬT TOÁN PHÁT SINH A, B, N TỪ MÃ BĂM MẬT KHẨU SHA256 32 BYTES

1. b = giá trị nguyên ngẫu nhiên 4 bytes
2. offset = 0
3. While a % n ==0:

```
a = hex_to_uint (LEFT(SHA256(H2), offset + 8), endian: ”big”)
a2 = hex_to_uint (LEFT(SHA256(H2), offset + 8), endian: ”little”)
n = largest_prime_not_divisior(a2)
if n is None:
        offset+=1
        n = 1
```

hex_to_uint: đổi 4 bytes hex ra số nguyên không dấu, với big (bit trọng số nhỏ giá trị nhỏ) hay little (bit trọng số nhỏ giá trị lớn)

largest_prime_not_divisior(a2): tìm số nguyên tố n lớn nhất của a2 mà a2 không chia hết cho n.

IV. KẾT QUẢ VÀ THẢO LUẬN

Giải pháp đề xuất đã tập trung cải tiến độ bảo mật của khóa công khai và tăng tính xác thực trong tiến trình trao đổi khóa bằng Diffie Hellman cải biên - một biến thể Diffie Hellman ứng dụng phục vụ xác thực nhẹ, giúp giảm thiểu rủi ro tấn công lượng tử và tấn công trung gian. Giải pháp không nhằm thay thế giao thức trao đổi khóa chuẩn, mà chỉ bổ sung thêm tính xác thực cho Diffie Hellman, đồng thời tập trung tối ưu hóa quy trình trao đổi dữ liệu để cải thiện hiệu năng, chống tấn công DDoS.

Kết quả thực nghiệm thu được từ việc triển khai ứng dụng trong môi trường client-server mô phỏng bằng code Python trên tập hình bmp và png tự sưu tầm cho mục đích nghiên cứu. Ứng dụng chạy trên máy tính CPU intel core I5, 2.6GHz, 4 core, 8 logic processors, 16GB RAM được trình bày trong bảng 2, bảng 3 và bảng 4. Các kết quả mang tính so sánh hiệu năng theo xu hướng kết hợp aes-256 cho mã hóa dữ liệu và thuật toán hậu lượng tử cho trao đổi khóa và chữ ký số.

Bảng 2. *So sánh hiệu năng tính toán SHA256, AES256, Kyber512, PBKDF2*

Thuật toán mã hóa	Thời gian	Số lần/giây
SHA256	0.33 ns	3,030,303,030
AES256+PBKDF2 (interation=1)	1.12 ms	935
Kyber512	8.29 ms	121
RSA 2048 bit	368 ms	2.7
Khối dữ liệu thực nghiệm: 32 bytes		

Bảng 3. *Thời gian thực hiện hoạt động xác thực tại server*

Thuật toán mã hóa	Thời gian	Số lần/giây
Xác thực với RSA 2048	351 ms	2.85
Xác thực với Kyber 512	9.08 ms	110.13
Xác thực với giải pháp để xuất	0.193 ms	5181.35
Dữ liệu xác thực thực nghiệm: 32 bytes không bao gồm thời gian truyền và nhận dữ liệu		

Bảng 4. *So sánh thời gian thực thi giải pháp tại server theo từng giai đoạn*

Giai đoạn	Thời gian	Tỉ lệ %
Xác thực client và mật khẩu user	0.193 ms	0.0002%
Tạo khóa chung passkey	401 ms	37.6879%
Trích xuất, giải mã dữ liệu truyền	128 ms	12.03%
Truyền dữ liệu qua mạng mô phỏng	535 ms	50.2819%
Tổng thời gian phiên giao dịch	1064 ms	
Dữ liệu xác thực thực nghiệm: thông điệp 32 bytes, file hình gốc png, kích thước 237KB, độ phân giải 670 x 386		
Với dữ liệu 12122 bytes, thời gian trích xuất và giải mã là 162 ms		

A. PHÂN TÍCH VỀ ĐỘ BẢO MẬT CỦA GIẢI PHÁP

Hạn chế rủi ro lộ mật khẩu khi mã băm lưu trữ trên server bị lấy cắp. Việc lưu trữ mật khẩu dưới dạng mã băm kép SHA1 và SHA256, giúp tăng khả năng chống va chạm nghịch ảnh, với số phép thử trung bình lên tới 2^{416}, vượt quá khả năng tính toán thực tế của máy tính silicon, cũng như máy tính lượng tử.

Ngăn chặn tấn công mật khẩu bằng bảng cầu vồng bằng việc sử dụng PBKDF2 với giá trị salt và interactions để tạo khóa mật K cho AES từ mã băm SHA256 của mật khẩu,

Tính bảo mật chuyển tiếp. Do việc tạo khóa bí mật chung passkey theo từng phiên kết nối, thực hiện trực tiếp tại client và server, dựa trên mã băm kép mật khẩu người dùng. Ngay cả khi mã băm kép mật khẩu bị lộ trong tương lai, hacker cũng không thể dùng thông tin này để giải mã dữ liệu các phiên đã thu thập trong quá khứ. Tấn công phiên hay cookie, không giúp hacker giả mạo được ứng dụng client, do passkey không được lưu trong session hay cookie, ngăn chặn khả năng vượt qua cơ chế xác thực đa lớp mà không cần mật khẩu của người dùng.

Giảm rủi ro tấn công trung gian do không có sự trao đổi khóa công khai trong Diffie Hellman cải biên nên không thể giả mạo khóa công khai trong quá trình kết nối, hacker cũng không thể tính được passkey để giải mã dữ liệu truyền trong suốt phiên giao dịch, mặt khác tiến trình xác thực còn kiểm tra serial number của client mỗi lần đăng nhập, giúp hạn chế khả năng đăng nhập ứng dụng server từ các thiết bị lạ.

Giải pháp có khả năng chống tấn công lượng tử do Diffie Hellman cải biên không trao đổi khóa công khai, nên ngăn chặn rủi ro tấn công lượng tử bài toán logarit rời rạc. Cơ chế phát sinh khóa công khai (a, n) và khóa bí mật (b) trong Diffie Hellman cải biên cũng không dựa trên việc phân tích số nguyên lớn thành thừa số nguyên tố như RSA hay ECC, nên không bị tác động bởi năng lực lượng tử trong tương lai. Việc bẻ khóa K hay passkey của AES256 nhằm xác định giá trị băm H1 hay dữ liệu truyền là bất khả thi. Vì ngoài khả năng chống tấn công lượng tử của AES256, hacker còn bị giới hạn bởi thời gian tấn công phiên và không biết mã băm H1, H2 để so khớp.

Hạn chế tấn công từ chối dịch vụ DDoS. Tiến trình phát sinh khóa bí mật chung passkey chỉ diễn ra khi tiến trình xác thực mật khẩu hợp lệ. Nên khi máy chủ nhận hàng loạt yêu cầu thiết lập kết nối giả mạo, thời gian xác thực các kết nối giả này là không đáng kể (bảng 4), do không cần phải thực hiện các phép tính khóa chung passkey tốn kém tài nguyên như Diffie Hellman, tránh làm quá tải server.

Không thể phát hiện được sự tồn tại của dữ liệu truyền, khi không xác định được giá trị seed của phiên kết nối, vì không thể bẻ khóa passkey của AES256 trong thời gian ngắn.

Khó trích xuất thông tin trong ảnh, ngay cả khi biết ảnh có ẩn giấu thông tin. Do mỗi thông điệp có thể sử dụng một giá trị seed và tệp hình gốc khác nhau, nên việc tấn công brute force giá trị seed để trích xuất thông điệp trong hình là không khả thi với các thông điệp bị mã hóa và số lượng phép thử lớn 2^{32}.

B. PHÂN TÍCH VỀ HIỆU NĂNG CỦA GIẢI PHÁP

Kết quả thực nghiệm trong bảng 2 cho thấy thời gian băm với SHA256 rất ngắn, nên việc tăng thêm một số lần băm trong quá trình xác thực không ảnh hưởng lớn đến hiệu suất hệ thống.

Từ kết quả thực nghiệm bảng 3 cho thấy giải pháp đề xuất có hiệu năng xác thực tại máy chủ tốt hơn các giải pháp xác thực dùng khóa công khai RSA/ECC, cũng như các giải pháp hậu lượng tử như Kyber.

Cải tiến hiệu năng tiến trình xác thực nhờ kích thước khóa và các tham số nhỏ hơn Diffie Hellman, với (a, b, n) 32 bit và passkey 256 bit, so với Diffie Hellman 3072bit hay RSA 2048 bit. Việc gia tăng số lần băm interations trong

PBKDF2 để chống tấn công Rainbow mật khẩu, không làm giảm hiệu năng của máy chủ khi xác thực, do thời gian thực hiện băm 32 bytes bằng SHA256 rất ngắn (bảng 2).

Từ kết quả thực nghiệm bảng 4 cho thấy việc tích hợp giải pháp giấu tin và mã hóa dữ liệu truyền tải, gia tăng đáng kể độ bảo mật của dữ liệu truyền, tuy có làm tăng tải xử lý server, nhưng vẫn nhanh hơn so với việc mã hóa RSA 2048 bit với cùng khối lượng dữ liệu truyền.

Ngoài việc tăng cường độ an toàn cho tiến trình trao đổi khóa hay truyền dữ liệu, giải pháp đề xuất đặc biệt hiệu quả trong trường hợp xác thực mật khẩu hoặc thực thi các giao dịch thanh toán trong thương mại điện tử, đòi hỏi tính bảo mật cao, dung lượng dữ liệu truyền ít trong mỗi phiên giao dịch.

V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Mục đích chính của nghiên cứu là cung cấp một cái nhìn tổng quan về các giải pháp trao đổi khóa hậu lượng tử. Từ đó đề xuất một giải pháp trao đổi khóa và truyền dữ liệu an toàn, hạn chế rủi ro tấn công lượng tử và tấn công trung gian, có chú ý đến việc tối ưu hóa quy trình trao đổi dữ liệu để cải thiện hiệu năng và hạn chế rủi ro tấn công DDoS.

Giải pháp đề xuất cho phép tăng cường độ an toàn cho tiến trình truyền dữ liệu trong mô hình thanh toán thương mại điện tử, sử dụng mã hóa AES để bảo mật nội dung, kết hợp kỹ thuật giấu tin để che giấu sự tồn tại của dữ liệu, với tốc độ xử lý nhanh hơn mã hóa RSA 2048 bit.

Giải pháp có thể dễ dàng triển khai trong thực tế trên mọi thiết bị người dùng mà không cần các thiết bị đặc thù có chức năng sinh trắc học hay tốn kém chi phí phát hành chứng chỉ số để đảm bảo tính xác thực. Phù hợp triển khai trong các mô hình ứng dụng bảo mật client-server qui mô nhỏ, theo xu hướng chuyển đổi các giao thức trao đổi khóa chuẩn trong giai đoạn hậu lượng tử.

Trong tương lai, tác giả sẽ tiếp tục nghiên cứu mở rộng, tích hợp các giải pháp xác thực đa lớp và zero trust, nhằm nâng cao hơn nữa tính xác thực từ cả hai phía client và server, để hạn chế rủi ro khi mã băm kép mật khẩu người dùng bị lộ, cũng như cải thiện hơn nữa hiệu năng truyền tải của giải pháp bằng cơ chế chọn hình có kích thước phù hợp với độ dài payload cần truyền tải.

VI. TÀI LIỆU THAM KHẢO

[1] L. Zhang, H. Liu (2025). Client-Server Computing in Modern E-Commerce Systems: architecture, performance, and security analysis, *Journal of Information Systems and E-Business Management*, Advance online publication, doi: 10.1007/s10257-025-0582-1

[2] Saru Kumari, Mtudul Dixit, Xiong Li, Fan Wu, Soumya Das (2016). Analysis and improvement of a key exchange and authentication protocol in client-server environment, *Wireless Personal Communications*, Vol. 86, No. 4, pp.2163-2187, doi: 10.1007/s11277-015-3004-9

[3] Gupta, R., & Sharma, N. (2025). Quantum threats to public-key infrastructure, *Journal of Information Security and Applications*, No. 75, pp.45–60, doi: 10.1016/j.jisa.2025.103456

[4] J. Chen, L. Wang (2024). Quantum Attacks on Diffie–Hellman Key Exchange, *IACR ePrint Archive*, Report 2024/567

[5] Dustin Moody Ray Perlner Andrew Regenscheid Angela Robinson David Cooper (2024, November): Transition to Post-Quantum Cryptography Standards, *NIST Internal Report*, *IR 8547 ipd*, doi: 10.6028/NIST.IR.8547.ipd

[6] B. Owolabi, B. Elly (2025). Optimizing Hybrid Key Exchange Protocols for Post-Quantum Security, *ResearchGate*, doi: 10.13140/RG.2.2.14659 .31525

[7] Nguyen, T. H., & Pham, Q. A. (2024). Developing a passwordless authentication solution using asymmetric cryptography and device fingerprint technology*, International Conference on Advanced Technologies for Communications (ATC)*, 1–6. IEEE, doi: 10.1109/ATC60774.2024.11009081

[8] Bhoi, S. (2024). Password strengthening: Using multi-lingual passwords, *International Journal of Innovative Science and Research Technology (IJISRT)*, 9(5), pp.1746–1750. Available: https://ijisrt.com/assets/upload/files/IJISRT24MAY1746.pdf

[9] Song, J., Han, J., Au, M. H., Yang, R., & Sun, C. (2025). Lattice-based dynamic k-times anonymous authentication with attribute-based credentials*, arXiv preprint arXiv:2509.21786*. Available: https://arxiv.org/abs/2509.21786

[10] NIST (2024, August 13), Module-Lattice-Based Key-Encapsulation Mechanism Standard, FIPS 203, doi: 10.6028/NIST.FIPS.203

[11] Check Point Research Team. (2024). Attackers find your session cookies irresistible*, Check Point Blog*. Available: https://blog.checkpoint.com/security/attackers-find-your-session-cookies-irresistible/

AN AUTHENTICATION AND DATA TRANSMISSION SECURIY SOLUTION AGAINST QUANTUM AND MAN-IN-THE-MIDDLE ATTACKS

Dinh Xuan Lam

ABSTRACT— Secure key exchange mechanisms remain a critical prerequisite for ensuring authentication and security in client-server connections. Traditional data encryption techniques based on PKI architecture are no longer an absolutely secure solution for key exchange and data transmission over networks, especially with the advent of quantum computers. This paper analyzes limitations and quantum attack risks in key exchange protocols based on public-key cryptosystems such as Diffie-Hellman. It then proposes a key exchange solution following an AES encryption pipeline model, incorporating an authenticated modified Diffie-Hellman scheme combined with steganography. The approach is capable of resisting man-in-the-middle attacks and mitigating security risks posed by quantum computing, thereby reducing the likelihood of information leakage or forgery and enhancing the safety of authentication and data transmission processes in client-server applications. The study also provides an overview of the performance and practical deployability of the solution based on experimental results.

Keywords— Client-server security, Steganography, Key exchange protocol, Man-in-the-middle attack prevention, Quantum-resistant communication.

Đinh Xuân Lâm là thạc sĩ Truyền dữ liệu và mạng máy tính Học viện Công nghệ Bưu chính Viễn thông TPHCM, giảng viên khoa Công nghệ thông tin, Trường Đại học Ngoại ngữ-Tin học TP.HCM. Lĩnh vực nghiên cứu quan tâm: An ninh mạng và quản trị hệ thống.